

Computing the Exponential of Large Block-Triangular Block-Toeplitz Matrices Encountered in Fluid Queues

D.A. Bini, Università di Pisa
S. Dendievel, Université libre de Bruxelles
G. Latouche, Université libre de Bruxelles
B. Meini, Università di Pisa

February 27, 2015

Abstract

The Erlangian approximation of Markovian fluid queues leads to the problem of computing the matrix exponential of a subgenerator having a block-triangular, block-Toeplitz structure. To this end, we propose some algorithms which exploit the Toeplitz structure and the properties of generators. Such algorithms allow to compute the exponential of very large matrices, which would otherwise be untreatable with standard methods. We also prove interesting decay properties of the exponential of a generator having a block-triangular, block-Toeplitz structure.

Keyword Matrix exponential, Toeplitz matrix, circulant matrix, Markov generator, fluid queue, Erlang approximation.

1 Introduction

The problem we consider here is to compute the exponential of an upper block-triangular, block-Toeplitz matrix, that is, a matrix of the kind

$$\mathcal{T}(U) = \begin{bmatrix} U_0 & U_1 & \dots & U_{n-1} \\ & U_0 & \ddots & \vdots \\ & & \ddots & U_1 \\ 0 & & & U_0 \end{bmatrix}, \quad (1)$$

where U_i , $i = 0, \dots, n-1$, are $m \times m$ matrices. Our interest stems from the analysis in Dendievel and Latouche [10] of the Erlangization method for Markovian fluid models, but the story goes further back in time.

1.1 Origin of the problem

The Erlangian approximation method was introduced in Asmussen *et al.* [2] in the context of risk processes; it was picked up in Stanford *et al.* [18] where a connection is established with fluid queues. Other relevant references are Stanford *et al.* [19] where the focus is on modelling the spread of forest fires, and Ramaswami *et al.* [16] where some basic algorithms are developed.

Markovian fluid models are two-dimensional processes $\{(X(t), \varphi(t)) : t \in \mathbb{R}^+\}$ where $\{\varphi(t)\}$ is a Markov process with infinitesimal generator A on the state space $\{1, \dots, m\}$; to each state i is associated a rate of growth $c_i \in \mathbb{R}$ and $X(t)$ is controlled by $\varphi(t)$ through the equation

$$X(t) = X(0) + \int_0^t c_{\varphi(s)} ds, \quad \text{for } t \geq 0.$$

Performance measures of interest include the distributions of $X(t)$ and of various first passage times. Usually, $\varphi(t)$ is called the phase of the process at time t and $X(t)$ its level, and the phase space $\{1, \dots, m\}$ is partitioned into three subsets \mathcal{S}_+ , \mathcal{S}_- and \mathcal{S}_0 such that $c_i > 0$, $c_i < 0$ or $c_i = 0$ if i is in \mathcal{S}_+ , \mathcal{S}_- or \mathcal{S}_0 , respectively. To simplify our presentation without missing any important feature, we assume below that \mathcal{S}_0 is empty.

The first return probabilities of $X(t)$ to its initial level $X(0)$ play a central role in the analysis of fluid queues. It is customary to define two matrices Ψ and $\hat{\Psi}$ of first return probabilities:

$$\Psi_{ij} = \Pr[\tau < \infty, \varphi(\tau) = j | X(0) = 0, \varphi(0) = i], \quad i \in \mathcal{S}_+, j \in \mathcal{S}_-,$$

and

$$\hat{\Psi}_{ij} = \Pr[\tau < \infty, \varphi(\tau) = j | X(0) = 0, \varphi(0) = i], \quad i \in \mathcal{S}_-, j \in \mathcal{S}_+,$$

where $\tau = \inf\{t > 0 : X(t) = 0\}$ is the first passage time to level 0. Thus, the entries of Ψ and $\hat{\Psi}$ are the probability of returning to the initial level after having started in the upward, and the downward directions, respectively.

If the process starts from some level $x > 0$, then

$$\Pr[\tau < \infty, \varphi(\tau) = j | X(0) = x, \varphi(0) = i] = \left(\begin{bmatrix} I \\ \Psi \end{bmatrix} e^{Hx} \right)_{ij} \quad i \in \{1, \dots, m\}, j \in \mathcal{S}_-;$$

here, H is a square matrix on $\mathcal{S}_- \times \mathcal{S}_-$ and is given by

$$H = |C_-|^{-1} A_{--} + |C_-|^{-1} A_{-+} \Psi,$$

where A_{--} and A_{-+} are submatrices of the generator A , indexed by $\mathcal{S}_- \times \mathcal{S}_-$ and $\mathcal{S}_- \times \mathcal{S}_+$, respectively, and $|C_-|$ is a diagonal matrix with $|c_i|, i \in \mathcal{S}_-$ on the diagonal. A similar equation holds for $x < 0$. The matrices Ψ and $\hat{\Psi}$ are solutions of algebraic Riccati equations and their resolution has been the object of much attention. Very efficient algorithms are available, and we refer to Bini *et al.* [7] and Bean *et al.* [3].

The Erlangian approximation method is introduced in [2] to determine the detailed distribution of τ . The idea is that, to compute the probability

$$F(t_0; i, x) = \Pr[\tau < t_0 | X(0) = x, \varphi(0) = i], \quad x > 0, i \in \{1, \dots, m\}$$

for a fixed value t_0 , it is convenient to replace t_0 by a random variable T with an Erlang distribution, with parameters $(n/t_0, n)$ for some positive integer n . The random variable T has expectation t_0 and variance t_0/n , so that $F(T; i, x)$ is a good approximation of $F(t_0; i, x)$ if n is large enough. From a computational point of view, the advantage is that one replaces systems of integro-differential equations by linear equations.

The long and the short of it is that the original system is replaced by the process $\{(X(t), \Phi(t))\}$ with a two-dimensional phase $\Phi(t) = (\beta(t), \varphi(t))$ on the state space $\{1, \dots, n\} \times \{1, \dots, m\} \cup \{0\}$ and with the generator

$$Q = \begin{bmatrix} A - \nu I & \nu I & & 0 & 0 \\ & A - \nu I & \ddots & & \\ & & \ddots & \nu I & \\ & & & A - \nu I & \nu \mathbf{1} \\ 0 & & & 0 & 0 \end{bmatrix},$$

where $\nu = n/t_0$. The physical interpretation is that the absorbing state 0 is entered at the random time T , and the component β of the phase marks the progress of time towards T . Some authors (for instance [2, 18]) report that good approximations may be obtained with small values of n .

Because of the Toeplitz-like structure of Q , the matrices Ψ and H are both upper block-triangular block-Toeplitz and it is interesting to use the Toeplitz structure in order to reduce the cost when n is large. This is done in [16] for the matrix Ψ . Here we address the question of efficiently computing the exponential matrix e^{Hx} for a given value of x , where H has the structure of (1). We shall assume without loss of generality that $x = 1$.

1.2 Main results

We recall that the exponential function can be extended to a matrix variable by defining

$$e^X = \sum_{i=0}^{\infty} \frac{1}{i!} X^i. \quad (2)$$

For more details on the matrix exponential and more generally on matrix functions we refer the reader to Higham [11].

The matrix $\mathcal{T}(U)$ defined in (1) is of order nm and it may be huge, since a larger n leads to a better Erlangian approximation, while the size m of the blocks is generally small. The matrix $\mathcal{T}(U)$ is a subgenerator, i.e., it has negative diagonal entries, nonnegative off-diagonal entries, and the sum of the entries on each row is nonpositive.

Since block-triangular block-Toeplitz matrices are closed under matrix multiplication, it follows from (2) that the matrix exponential $e^{\mathcal{T}(U)}$ is also an upper block triangular, block-Toeplitz matrix; in particular, the diagonal blocks of $e^{\mathcal{T}(U)}$ coincide with e^{U_0} . Moreover, it is known that the matrix $e^{\mathcal{T}(U)}$ is nonnegative and substochastic.

The problem of the computation of the exponential of a generator has been considered in Xue and Ye [21, 20] and by Shao et al. [17], where the authors propose component-wise accurate algorithms for the computation. These algorithms are efficient for matrices of small size. For the Erlangian approximation problem, these algorithms are useless for the large size of the matrices involved. Recently, some attention has been given to the computation of the exponential of general Toeplitz matrices by using Arnoldi method (Lee *et al.* [13], Pang and Sun [15]).

In our framework, Toeplitz matrices are block-triangular so that they form a matrix algebra. This property is particularly effective for the design of efficient algorithms and we propose some numerical methods that exploit the block-triangular block-Toeplitz structure and the generator properties. Unlike the general methods, our algorithms allow one to deal with matrices $\mathcal{T}(U)$ of very large size.

Two methods rely on spectral and computational properties of block-circulant and block ϵ -circulant matrices (Bini [6], Bini *et al.* [8]) and on the use of Fast Fourier Transforms (FFT). Recall that block ϵ -circulant matrices have the form

$$\mathcal{C}_\epsilon(U) = \begin{bmatrix} U_0 & U_1 & \dots & U_{n-1} \\ \epsilon U_{n-1} & U_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & U_1 \\ \epsilon U_1 & \dots & \epsilon U_{n-1} & U_0 \end{bmatrix},$$

and that a block-circulant matrix is a block ϵ -circulant matrix with $\epsilon = 1$. For simplicity, we denote by $\mathcal{C}(U)$ the block 1-circulant matrix $\mathcal{C}_1(U)$.

Since block ϵ -circulant matrices can be block-diagonalized by FFT [6], the computation of the exponential of an $n \times n$ block ϵ -circulant matrix with $m \times m$ blocks can be reduced to the computation of n exponentials of $m \times m$ matrices. These latter exponentials are independent from each other and can be computed simultaneously with a multi-core architecture at the cost of a single exponential.

The idea of the first method is to approximate $e^{\mathcal{T}(U)}$ by $e^{\mathcal{C}_\epsilon(U)}$ where $\epsilon \in \mathbb{C}$ and $|\epsilon|$ is sufficiently small. We analyse the error and are thereby able to choose the value of ϵ which gives a good balance between the roundoff error and the approximation error. In fact, the approximation error grows as $O(\epsilon)$ while the roundoff error is $O(\mu\epsilon^{-1})$, where μ is the machine precision. This leads to an overall error which is $O(\mu^{1/2})$. By using the fact that the solution is real, by choosing ϵ a pure imaginary number we get an approximation error $O(\epsilon^2)$ which leads to an overall error $O(\mu^{2/3})$.

Since the approximation error is a power series in ϵ , we devise a further technique which consists in averaging the solutions computed with k different

values of ϵ . This way, we are able to cancel out the components of the error of degree less than ϵ^{2k} . This leads to a substantial improvement of the precision. Moreover, since the different computations are independent from each other, the computational cost in a multicore architecture is independent of k .

In our second approach, the matrix $\mathcal{T}(U)$ is embedded into a $K \times K$ block-circulant matrix $\mathcal{C}(U^{(K)})$, where K is sufficiently large, and an approximation of $e^{\mathcal{T}(U)}$ is obtained from a suitable submatrix of $e^{\mathcal{C}(U^{(K)})}$. The computation of $e^{\mathcal{C}(U^{(K)})}$ is reduced to the computation of K exponentials of $m \times m$ matrices, and our error analysis allows one to choose the value of K so as to guarantee a given error bound in the computed approximation.

The third numerical method consists in specializing the shifting and Taylor series method of [20]. The block-triangular Toeplitz structure is exploited in the FFT-based matrix multiplications involved in the algorithm, leading to a reduction of the computational cost. The algorithm obtained in this case does not seem well suited for an implementation in a multicore architecture.

We compare the three numerical methods, from a theoretical as well as from a numerical point of view. From our analysis, we conclude that the method based on ϵ -circulant matrices is the fastest and provides a reasonable approximation to the solution. Moreover, by applying the averaging technique we can dramatically improve the accuracy. The method based on embedding and the one based on power series perform an accurate computation but are slightly more expensive.

It must be emphasised that the use of FFT makes the algorithms norm-wise stable but that component-wise stability is not guaranteed. In consequence, the matrix elements with values of modulus below the machine precision may not be well approximated in terms of relative error.

The paper is organised as follows. In Sections 2 and 3, we recall properties of the exponential of a subgenerator and of its derivatives, and some basic properties of block-Toeplitz and block-circulant matrices which are used in our algorithms. In Section 4, we show how to compute the exponential of a block ϵ -circulant matrix by using fast arithmetic based on FFT and we perform an error analysis. We present in Section 5 the algorithms to compute the exponential of $\mathcal{T}(U)$: first we analyse the decay of off-diagonal entries of the matrix exponential, next we describe the new methods and perform an error analysis. We conclude with numerical experiments in Section 6.

2 The exponential of a subgenerator and its derivatives

2.1 The exponential of a subgenerator

A subgenerator of a Markov process is a matrix Q of real numbers such that the off-diagonal entries of Q are nonnegative, the diagonal entries are negative, and the sum of the entries on each row is nonpositive. We denote by $\mathbf{1}$ the column vector with all entries equal to 1, with size according to the context. If Q is a

subgenerator, then $Q\mathbf{1} \leq 0$ and Q is called a generator if the row sum on all rows is zero.

Let $\sigma = \max_i(-q_{ii})$. The matrix $V = Q + \sigma I$ is a nonnegative matrix, and we may write $e^Q = e^{V-\sigma I} = e^{-\sigma} e^V$. From the latter equality it follows that the matrix exponential e^Q is nonnegative. Moreover, since $Q\mathbf{1} \leq 0$ it follows that $V\mathbf{1} = Q\mathbf{1} + \sigma\mathbf{1} \leq \sigma\mathbf{1}$. Therefore, in view of (2), $e^Q\mathbf{1} = e^{-\sigma} e^V\mathbf{1} = e^{-\sigma} \sum_{i=0}^{\infty} \frac{1}{i!} V^i\mathbf{1} \leq e^{-\sigma} e^{\sigma}\mathbf{1}$. Thus we may conclude that $e^Q\mathbf{1} \leq \mathbf{1}$, that is, e^Q is a substochastic matrix.

2.2 Derivatives and perturbation results

We recall the definition and some properties of the Gâteaux and Fréchet derivatives, and their expression for the matrix exponential function, together with some properties when the matrix is a subgenerator. We refer the reader to [11] for more details.

The Fréchet derivative of a matrix function $f : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ at a point $X \in \mathbb{C}^{n \times n}$ along the direction $E \in \mathbb{C}^{n \times n}$ is the linear mapping $L(X, E)$ in the variable E such that

$$f(X + E) - f(X) - L(X, E) = o(\|E\|). \quad (3)$$

The Gâteaux (or directional) derivative of $f : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ at a point $X \in \mathbb{C}^{n \times n}$ along the direction $E \in \mathbb{C}^{n \times n}$ is

$$G(X, E) = \lim_{h \rightarrow 0} \frac{f(X + hE) - f(X)}{h}. \quad (4)$$

If the Fréchet derivative exists, then it is equal to the Gâteaux derivative ([11, Section 3.2]). Such is the case for the matrix exponential function and we may, therefore, use either definition (3) or (4), depending on which is more convenient; we will use the Gâteaux derivative. From [14],

$$G(tX, E) = \int_0^t e^{X(t-s)} E e^{Xs} ds \quad (5)$$

and the following equation gives an expression for the matrix exponential in terms of Gâteaux derivatives:

$$e^{t(X+hE)} = \sum_{j=0}^{\infty} \frac{h^j}{j!} G^{[j]}(tX, E) \quad (6)$$

where we denote by $G^{[j]}(tX, E)$ the j -th Gâteaux derivative of the matrix function e^{tX} in the direction E , obtained by the recurrence equation

$$G^{[j]}(tX, E) = j \int_0^t e^{(t-s)X} E G^{[j-1]}(sX, E) ds, \quad j = 1, 2, \dots, \quad (7)$$

and $G^{[0]}(tX, E) = e^{tX}$.

Recall that if X is a subgenerator, then e^{tX} is a substochastic matrix for any $t \geq 0$ and in particular $\|e^{tX}\|_\infty \leq 1$. Therefore, by taking norms in (5), we obtain the upper bound

$$\|G(tX, E)\|_\infty \leq \int_0^t \|e^{X(t-s)}\|_\infty \|E\|_\infty \|e^{Xs}\|_\infty ds \leq t\|E\|_\infty \quad (8)$$

which may be extended to the j -th order Gâteaux derivative as in the next proposition.

Proposition 1 *If X is a subgenerator, then*

$$\|G^{[j]}(tX, E)\|_\infty \leq t^j \|E\|_\infty^j \quad (9)$$

for $t \geq 0$, for any $j \geq 0$. Moreover, if E is a nonnegative matrix, then $G^{[j]}(tX, E)$ is nonnegative for any $j \geq 0$.

Proof. Since the matrix X is a subgenerator, the matrix $e^{\tau X}$ is nonnegative and substochastic for any $\tau \geq 0$, therefore $\|e^{\tau X}\|_\infty \leq 1$ for any $\tau \geq 0$. By using this property, the inequality (9) can be proved by induction. If $j = 0$, then $\|G^{[0]}(tX, E)\|_\infty \leq 1$. The inductive step is immediately proved, since from (7) we have

$$\begin{aligned} \|G^{[j]}(tX, E)\|_\infty &\leq j \int_0^t \|e^{(t-s)X}\|_\infty \|E\|_\infty \|G^{[j-1]}(sX, E)\|_\infty ds \\ &\leq j \int_0^t \|E\|_\infty^j s^{j-1} ds = \|E\|_\infty^j t^j, \end{aligned}$$

where the last inequality follows from the inductive assumption. If the matrix E is nonnegative, from the recurrence (7) and from the fact that $e^{\tau X}$ is nonnegative and substochastic for any $\tau \geq 0$, it follows by induction that $G^{[j]}(tX, E)$ is nonnegative for any $j \geq 0$. \square

The following result provides some bounds related to the exponential of the matrix $\mathcal{T}(U)$ of (1) and to its Gâteaux derivative; it will be used in the next sections to analyse the stability of the algorithm in Section 5.2 based on ϵ -circulant matrices.

Theorem 2 *Let $\mathcal{T}(U)$ be the matrix in (1) and assume it is a subgenerator. For $x = (x_i)_{i=1, \dots, n-1} \in \mathbb{C}^{n-1}$ define $H(x) = U_0 + \sum_{i=1}^{n-1} x_i U_i$. If $|x_i| \leq 1$, $i = 1, \dots, n-1$, then $\|e^{sH(x)}\|_\infty \leq 1$ for any $s \geq 0$. Moreover, $\|G(H(x), E)\|_\infty \leq \|E\|_\infty$ for any $m \times m$ matrix E .*

Proof. Define $\alpha = \max_i (-(U_0)_{ii})$, $\tilde{H} = H((1, \dots, 1))$ and $B = \tilde{H} + \alpha I$. From the choice of α it follows that $B \geq 0$. We have

$$\|e^{s\tilde{H}}\|_\infty = \|e^{sB - s\alpha I}\|_\infty = e^{-s\alpha} \|e^{sB}\|_\infty \leq e^{-s\alpha} e^{\|sB\|_\infty}$$

where the latter inequality holds in view of [11, Theorem 10.10]. Since $B \geq 0$ we may write that $\|B\|_\infty = \|B\mathbf{1}\|_\infty$. From the inequality $(\sum_{k=0}^{n-1} U_k)\mathbf{1} \leq 0$ we find

that $(\alpha I + \sum_{k=0}^{n-1} U_k)\mathbf{1} \leq \alpha\mathbf{1}$, that is $B\mathbf{1} \leq \alpha\mathbf{1}$ whence $\|sB\|_\infty \leq s\alpha$. Therefore we conclude that $\|e^{s\tilde{H}}\|_\infty \leq 1$ and the first claim is proved.

Now, concerning $H(x)$ we have

$$e^{sH(x)} = e^{sH(x)+s\alpha I-s\alpha I} = e^{-s\alpha} e^{sH(x)+s\alpha I}. \quad (10)$$

Since $|x_k| \leq 1$, $U_0 + \alpha I \geq 0$ and $U_k \geq 0$, $k = 1, \dots, n-1$, we have

$$|H(x) + \alpha I| \leq |U_0 + \alpha I| + \left| \sum_{k=1}^{n-1} x_k U_k \right| \leq U_0 + \alpha I + \sum_{k=1}^{n-1} U_k = B.$$

By monotonicity of the infinity norm, we have $\|H(x) + \alpha I\|_\infty \leq \|B\|_\infty$,

$$\|e^{sH(x)+s\alpha I}\|_\infty \leq e^{\|sH(x)+s\alpha I\|_\infty} \leq e^{\|sB\|_\infty} \leq e^{s\alpha},$$

and, from (10), $\|e^{sH(x)}\|_\infty = e^{-s\alpha} \|e^{sH(x)+s\alpha I}\|_\infty \leq 1$. From (5), we have

$$\|G(H(x), E)\|_\infty \leq \|E\|_\infty \int_0^1 \|e^{(1-s)H(x)}\|_\infty \|e^{sH(x)}\|_\infty ds \leq \|E\|_\infty$$

and the last claim follows. \square

3 Fast computations with Toeplitz and circulant matrices

In this section we recall some basic properties of block-Toeplitz and block-circulant matrices, useful for our computational analysis. We refer the reader to Bini and Pan [9] and Bini *et al.* [8] for more details. Given a matrix $V \in \mathbb{C}^{m \times n}$, we denote by V^T and by V^H the transpose matrix and the transpose conjugate matrix of V , respectively. The conjugate of a complex number z is denoted by \bar{z} .

Let \mathbf{i} be the imaginary unit such that $\mathbf{i}^2 = -1$ and $\omega_n = \cos \frac{2\pi}{n} + \mathbf{i} \sin \frac{2\pi}{n}$ be a primitive n th root of the unity. We denote by $F = (\omega_n^{ij})_{i,j=0,n-1}$ the Fourier matrix. Recall that F is nonsingular, that $F^{-1} = \frac{1}{n} F^H$ and that, given a vector $v \in \mathbb{C}^n$, the application $v \rightarrow u = Fv$ defines the inverse discrete Fourier transform (IDFT) of v . We assume that n is an integer power of 2, so that the vector u can be computed by means of the FFT algorithm in $\frac{3}{2}n \log_2 n$ arithmetic operations (ops). The application $u \rightarrow v = \frac{1}{n} F^H u$ is called Discrete Fourier Transform (DFT) and the vector v can be computed in $\frac{3}{2}n \log_2 n + n$ ops.

Given the $m \times m$ matrices V_i , $i = 0, \dots, n-1$, we denote by $V = (V_i)_{i=0,n-1}$ the block-(column) vector with block-entries V_i , $i = 0, \dots, n-1$. Finally, we define $\mathcal{F} = F \otimes I_m$, where \otimes is the Kronecker product and I_m the identity matrix of order m . This way, for a block-column vector V the matrix $U = \mathcal{F}V$ can be computed by means of m^2 IDFTs with $\frac{3}{2}nm^2 \log_2 n$ ops. Similarly, given the matrix U , the block-vector $V = \frac{1}{n} \mathcal{F}^H U$ can be computed with $\frac{3}{2}nm^2 \log_2 n + nm^2$ ops.

3.1 Block-circulant matrices

For the results in this section we refer the reader to the book [9] and to the references cited therein.

Given the block-vector $U = (U_i)_{i=0, n-1}$, with $m \times m$ blocks, the $n \times n$ block-circulant matrix $\mathcal{C}(U) = (C_{i,j})_{i,j=0, n-1}$ associated with U is the matrix with block-entries

$$C_{i,j} = U_{j-i \bmod n}$$

so that $[U_0, \dots, U_{n-1}]$ coincides with the first block-row of $\mathcal{C}(U)$ and the entries of any other block-row are obtained by the entries of the previous block-row by a cyclic permutation which moves the last block entry to the first position and shifts the remaining block-entries one place to the right. For instance, for $n = 4$ one has

$$\mathcal{C}(U) = \begin{bmatrix} U_0 & U_1 & U_2 & U_3 \\ U_3 & U_0 & U_1 & U_2 \\ U_2 & U_3 & U_0 & U_1 \\ U_1 & U_2 & U_3 & U_0 \end{bmatrix}.$$

Observe that a block-circulant matrix is a particular block-Toeplitz matrix.

Block-circulant matrices can be simultaneously block-diagonalized by means of FFT, that is,

$$\frac{1}{n} \mathcal{F}^H \mathcal{C}(U) \mathcal{F} = \text{diag}(V_0, \dots, V_{n-1}), \quad V = \mathcal{F}U.$$

This property shows that block-circulant matrices are closed under matrix multiplication, i.e., they form a matrix algebra, moreover the product of a circulant matrix and a vector can be computed by means of Algorithm 1. This algorithm performs the computation with $2m^2$ FFTs and n matrix multiplications. Since $2m^3 - m^2$ ops are sufficient to multiply two $m \times m$ matrices, the overall cost of Algorithm 1 is $3nm^2 \log_2 n + nm^2 + (2m^3 - m^2)n$ ops.

Algorithm 1: Product of a block-circulant matrix and a block-vector

Input : Two block-vectors $X = (X_i)_{i=0, n-1}$, $U = (U_i)_{i=0, n-1}$

Output: The block-vector $Y = \mathcal{C}(U)X$, $Y = (Y_i)_{i=0, n-1}$

- 1 $V = \mathcal{F}U$
 - 2 $Z = \mathcal{F}X$
 - 3 $W_i = V_i Z_i, i = 0, \dots, n-1$, $W = (W_i)_{i=0, n-1}$
 - 4 $Y = \frac{1}{n} \mathcal{F}^H W$
-

If the input block-vectors are real then the vectors $V = \mathcal{F}U$ and $Z = \mathcal{F}X$ have a special structure, that is, the components V_0, Z_0 and $V_{n/2}, Z_{n/2}$ are real while $V_i = \bar{V}_{n-i}$, $Z_i = \bar{Z}_{n-i}$, for $i = 1, \dots, n/2 - 1$. In this case, the number of matrix multiplications at step 3 of Algorithm 1 is reduced to $n/2$.

Remark 3 Observe that the product of two circulant matrices may be computed by means of a product of a circulant matrix and a vector by means of Algorithm 1. In fact, since the last column of the block-circulant matrix $\mathcal{C}(U)$ is the block-vector $\hat{U} = (U_{n-i-1})_{i=0, n-1}$, if $\mathcal{C}(Y) = \mathcal{C}(U)\mathcal{C}(X)$ then we find that $\hat{Y} = \mathcal{C}(U)\hat{X}$, where $\hat{X} = (X_{n-i-1})_{i=0, n-1}$, $\hat{Y} = (Y_{n-i-1})_{i=0, n-1}$.

3.2 Block-triangular Toeplitz matrices

We denote by U the block-vector $(U_i)_{i=0, n-1}$ and by $\mathcal{T}(U)$ the block-upper triangular block-Toeplitz matrix whose first row is $[U_0, \dots, U_{n-1}]$. For $n = 4$, for instance,

$$\mathcal{T}(U) = \begin{bmatrix} U_0 & U_1 & U_2 & U_3 \\ 0 & U_0 & U_1 & U_2 \\ 0 & 0 & U_0 & U_1 \\ 0 & 0 & 0 & U_0 \end{bmatrix}.$$

Block-upper triangular block-Toeplitz matrices are closed under matrix multiplication.

Consider the vector \tilde{U} of $2n$ components obtained by filling the vector U with zero blocks, and the block-vector $V = (V_i)_{i=0, n-1}$ such that $V_0 = 0$ and $V_i = U_{n-i}$ for $i = 1, \dots, n-1$. Then the matrix $\mathcal{C}(\tilde{U})$ can be partitioned as follows

$$\mathcal{C}(\tilde{U}) = \begin{bmatrix} \mathcal{T}(U) & \mathcal{L}(V) \\ \mathcal{L}(V) & \mathcal{T}(U) \end{bmatrix}, \quad (11)$$

where $\mathcal{L}(V)$ is the block-lower triangular block-Toeplitz matrix whose first block-column is V . This expression enables one to compute the product $Y = \mathcal{T}(U)X$ of a block-upper triangular Toeplitz matrix and a block-vector with a low number of arithmetic operations. In fact, from (11) one deduces that Y coincides with the first half of the block-vector $\tilde{Y} = \mathcal{C}(\tilde{U})\tilde{X}$ where \tilde{X} is the block-vector of length $2n$ obtained by filling X with zeros. This fact leads to Algorithm 2 for computing the product of a block-triangular block-Toeplitz matrix and a block-vector. The cost of this algorithm is $6nm^2 \log_2(2n) + 2nm^2 + 2(2m^3 - m^2)n$ ops.

Algorithm 2: Product of a block-triangular block-Toeplitz matrix and a block-vector

Input : Two block-vectors $X = (X_i)_{i=0, n-1}$, $U = (U_i)_{i=0, n-1}$

Output: The block-vector $Y = \mathcal{T}(U)X$, $Y = (Y_i)_{i=0, n-1}$

- 1 Set $\tilde{X} = (\tilde{X}_i)_{i=0, 2n-1}$ with $\tilde{X}_i = X_i$ for $i = 0, \dots, n-1$, $\tilde{X}_i = 0$ for $i = n, \dots, 2n-1$
 - 2 Set $\tilde{U} = (\tilde{U}_i)$ with $\tilde{U}_i = U_i$ for $i = 0, \dots, n-1$, $\tilde{U}_i = 0$ for $i = n, \dots, 2n-1$
 - 3 Apply Algorithm 1 to compute $\tilde{Y} = \mathcal{C}(\tilde{U})\tilde{X}$
 - 4 Set $Y = (Y_i)_{i=0, n-1}$ with $Y_i = \tilde{Y}_i$, for $i = 0, \dots, n-1$.
-

Remark 4 Observe that the product of two block-triangular block-Toeplitz matrices can be computed by means of a product of a block-triangular block-Toeplitz matrix and a block-vector by means of Algorithm 2. In fact, since the last column of $\mathcal{T}(U)$ is the block vector $\widehat{U} = (U_{n-i-1})_{i=0, n-1}$, if $\mathcal{T}(Y) = \mathcal{T}(U)\mathcal{T}(X)$ then we find that $\widehat{Y} = \mathcal{T}(U)\widehat{X}$, where $\widehat{X} = (X_{n-i-1})_{i=0, n-1}$, $\widehat{Y} = (Y_{n-i-1})_{i=0, n-1}$.

3.3 Block- ϵ -circulant matrices

Given a block-vector U and a complex number ϵ , the block- ϵ -circulant matrix $\mathcal{C}_\epsilon(U) = (C_{i,j})$ is defined by

$$C_{i,j} = \begin{cases} U_{j-i} & \text{for } j \geq i, \\ \epsilon U_{n+j-i} & \text{for } j < i. \end{cases}$$

For instance, for $n = 4$ one has

$$\mathcal{C}_\epsilon(U) = \begin{bmatrix} U_0 & U_1 & U_2 & U_3 \\ \epsilon U_3 & U_0 & U_1 & U_2 \\ \epsilon U_2 & \epsilon U_3 & U_0 & U_1 \\ \epsilon U_1 & \epsilon U_2 & \epsilon U_3 & U_0 \end{bmatrix}.$$

Observe that a block- ϵ -circulant matrix is a particular case of block-Toeplitz matrix and that, for $|\epsilon|$ small, a block ϵ -circulant matrix is an approximation of a block-triangular block-Toeplitz matrix.

Like block-circulant matrices, block- ϵ -circulant matrices can be simultaneously block-diagonalized by means of FFT, so that they are closed under matrix multiplication and form a matrix algebra as well. In fact, one can show that

$$\frac{1}{n} \mathcal{F} \mathcal{D}_\epsilon^{-1} \mathcal{C}_\epsilon(U) \mathcal{D}_\epsilon \mathcal{F}^H = \text{diag}(V_0, \dots, V_{n-1}), \quad V = \mathcal{F}^H \mathcal{D}_\epsilon U, \quad (12)$$

where

$$\mathcal{D}_\epsilon = D_\epsilon \otimes I_m, \quad D_\epsilon = \text{diag}(1, \theta, \theta^2, \dots, \theta^{n-1}), \quad \theta = \epsilon^{1/n}.$$

4 The exponential of a block- ϵ -circulant matrix

Let $U = (U_i)_{i=0, n-1}$ be a block-vector of length n where $U_i \in \mathbb{C}^{m \times m}$, consider the block- ϵ -circulant matrix $\mathcal{C}_\epsilon(U)$ and its matrix exponential $e^{\mathcal{C}_\epsilon(U)}$. In view of (12), we find that

$$e^{\mathcal{C}_\epsilon(U)} = \frac{1}{n} \mathcal{D}_\epsilon \mathcal{F}^H \text{diag}(e^{V_0}, \dots, e^{V_{n-1}}) \mathcal{F} \mathcal{D}_\epsilon^{-1}, \quad V = \mathcal{F}^H \mathcal{D}_\epsilon U.$$

Therefore the exponential of a block- ϵ -circulant matrix is still block- ϵ -circulant. Moreover, we have $e^{\mathcal{C}_\epsilon(U)} = \mathcal{C}_\epsilon(Y)$ where

$$Y = \frac{1}{n} \mathcal{D}_\epsilon^{-1} \mathcal{F} W, \quad W = (W_i)_{i=0, n-1}, \quad W_i = e^{V_i}, \quad i = 0, \dots, n-1, \quad (13)$$

and $V = \mathcal{F}^H \mathcal{D}_\epsilon U$. The above equations allow to compute the exponential of an $n \times n$ block- ϵ -circulant matrix by computing n exponentials of $m \times m$ matrices and two Fourier transforms, as described in Algorithm 3.

Algorithm 3: Exponential of a block- ϵ -circulant matrix

Input : A complex number ϵ , the block-vector $U = (U_i)_{i=0, n-1}$ defining the first block-row of the ϵ -circulant matrix $\mathcal{C}_\epsilon(U)$

Output : The block-vector $Y = (Y_i)_{i=0, n-1}$ such that $\mathcal{C}_\epsilon(Y) = e^{\mathcal{C}_\epsilon(U)}$

- 1 $Z = \mathcal{D}_\epsilon U$
 - 2 $V = \mathcal{F}^H Z$
 - 3 $W_i = e^{V_i}, i = 0, \dots, n-1$, and set $W = (W_i)_{i=0, n-1}$
 - 4 $R = \frac{1}{n} \mathcal{F} W$
 - 5 $Y = \mathcal{D}_\epsilon^{-1} R$
-

Observe that the multiplication of U by the diagonal matrix \mathcal{D}_ϵ at step 1 reduces to scaling the blocks U_i by the scalar θ_i . The multiplication by $\mathcal{D}_\epsilon^{-1}$ at step 5 performs similarly. Therefore the overall cost of the algorithm is given by $3m^2n \log_2 n + 3m^2n$ ops plus the cost of computing n exponentials of $m \times m$ matrices.

For $\epsilon = 1$ the block- ϵ -circulant matrix turns to a block-circulant matrix and Algorithm 3 takes the simpler form described in Algorithm 4. The computational cost in this case is reduced to $3m^2n \log_2 n + m^2n$ ops plus the cost of computing n exponentials of $m \times m$ matrices.

Algorithm 4: Exponential of a block-circulant matrix

Input : The block-vector $U = (U_i)_{i=0, n-1}$ defining the first block-row of $\mathcal{C}(U)$

Output : The block-vector $Y = (Y_i)_{i=0, n-1}$ such that $\mathcal{C}(Y) = e^{\mathcal{C}(U)}$

- 1 $V = \mathcal{F}^H U$
 - 2 $W_i = e^{V_i}, i = 0, \dots, n-1$, and set $W = (W_i)_{i=0, n-1}$
 - 3 $Y = \frac{1}{n} \mathcal{F} W$
-

4.1 Numerical stability

Let $U = (U_i)_{i=0, n-1}$ be the block-vector defining the first block row of the sub-generator $\mathcal{T}(U)$. We analyze the error generated by computing the exponential of the block- ϵ -circulant matrix $\mathcal{C}_\epsilon(U)$ by means of Algorithm 3 in floating point arithmetic, where $\epsilon \in \mathbb{C}$ with $|\epsilon| < 1$.

Here and hereafter $\text{fl}(\cdot)$ denotes the result computed in floating point arithmetic of the expression between parenthesis. The symbol \doteq denotes equality up

to lower order terms, and similarly the symbol $\dot{\leq}$ stands for inequality up to lower order terms. The symbol μ denotes the machine precision.

We recall the following useful fact (see [11, page 71])

$$\text{fl}(xy) = xy(1 + \eta), \quad |\eta| \leq \frac{2\sqrt{2}}{1 - 2\mu}\mu =: \beta\mu, \quad \beta \doteq 2\sqrt{2}, \quad (14)$$

for $x, y \in \mathbb{C}$, and we use the following properties involving norms, where $v \in \mathbb{C}^n$

$$\begin{aligned} \|v\|_\infty &\leq \|v\|_2, \quad \|v\|_2 \leq \sqrt{n}\|v\|_\infty, \quad \|v\|_2 \leq \|v\|_1, \\ \|Fv\|_\infty &\leq n\|v\|_\infty, \quad \|Fv\|_2 \leq \sqrt{n}\|v\|_2. \end{aligned} \quad (15)$$

In order to perform the error analysis of Algorithm 3, we recall the following result concerning FFT (see [11, page 453]).

Theorem 5 *Let x be a vector of n components, $n = 2^q$, q integer, $y = Fx$, where $F = (\omega_n^{ij})_{i,j=0,n-1}$ is the Fourier matrix. Let \tilde{y} be the vector obtained in place of y by applying the Cooley-Tukey FFT algorithm in floating point arithmetic with precision μ where the roots of the unity are approximated by floating point numbers up to the error ν . Then*

$$\frac{\|y - \tilde{y}\|_2}{\|y\|_2} \leq \frac{q\eta}{1 - q\eta}, \quad \eta = \nu + (\sqrt{2} + \nu)\frac{4\mu}{1 - 4\mu}.$$

In particular, with $\nu = \mu$ and performing a first-order error analysis where we consider only the part of the error which is linear in μ we have

$$\frac{\|y - \tilde{y}\|_2}{\|y\|_2} \leq \gamma\mu q, \quad \gamma \doteq 4\sqrt{2} + 1. \quad (16)$$

Observe that, since $F^H = \overline{F}$, we may replace F with F^H in the statement of Theorem 5.

We split Algorithm 3 into three parts. The first part consists in computing the entries of the matrices V_k by means of steps 1 and 2, the second part consists in computing the entries of $W_k = e^{V_k}$ and the third part is formed by the remaining steps 4 and 5. The first and third part can be viewed as the collection of m^2 independent computations applied to the entry (r, s) of the generic block for $r, s = 1, \dots, m$. More specifically, given the pair (r, s) , denote $u = (u_k)$, $z = (z_k)$, $v = (v_k) \in \mathbb{C}^n$ the vectors whose components are $(U_k)_{r,s}$, $(Z_k)_{r,s}$, $(V_k)_{r,s}$, $k = 0, \dots, n-1$, respectively. The computation of u is obtained in the following way: $\theta = \epsilon^{1/n}$, $z_k = \theta^k u_k$, for $k = 0, \dots, n-1$, $v = F^H z$. While, denoting $w, r, y \in \mathbb{C}^n$ the vectors whose components are $(W_k)_{r,s}$, $(R_k)_{r,s}$, $(Y_k)_{r,s}$, $k = 0, \dots, n-1$, respectively, the computation of y is obtained in the following way: $r = \frac{1}{n} F w$, $y_k = \theta^{-k} r_k$ for $k = 0, \dots, n-1$.

Define $\delta_z = \tilde{z} - z$, $\delta_v = \tilde{v} - v$, $\delta_r = \tilde{r} - r$, $\delta_y = \tilde{y} - y$ where $\tilde{z}, \tilde{v}, \tilde{r}, \tilde{y}$ are the values obtained in place of z, v, r, y by performing computations in floating point arithmetic. We denote also by $(\delta_r)_k = \tilde{r}_k - r_k$ and $(\delta_y)_k = \tilde{y}_k - y_k$ the k -th component of δ_r and δ_y , respectively.

In our analysis we assume that the constants θ^k have been precomputed and approximated with the numbers θ_k such that $\theta_k = \theta^k(1 + \sigma_k)$, $|\sigma_k| \leq \mu$, $k = -n+1, \dots, 0, \dots, n-1$.

Since $z_k = \theta^k u_k$, from (14) we find that $\text{fl}(\theta^k u_k) = \theta^k u_k(1 + \eta_k)(1 + \sigma_k) \doteq \theta^k u_k(1 + \eta_k + \sigma_k)$. Thus,

$$\|\delta_z\|_\infty \leq \mu\zeta\|z\|_\infty, \quad \zeta \doteq \beta + 1. \quad (17)$$

Denoting by δ' the error introduced in computing the FFT v of z in floating point arithmetic, we have

$$\delta_v = F^H \delta_z + \delta',$$

and in view of (15), (16), and (17) we obtain

$$\begin{aligned} \|\delta_v\|_\infty &\leq n\|\delta_z\|_\infty + \|\delta'\|_2 \leq n\|\delta_z\|_\infty + \mu\gamma \log_2 n \|v\|_2 \\ &\leq \mu\zeta n\|z\|_\infty + \mu\gamma n \log_2 n \|z\|_\infty \\ &= \mu n\|z\|_\infty (\zeta + \gamma \log_2 n) \\ &\leq \mu n\|u\|_\infty (\zeta + \gamma \log_2 n), \end{aligned}$$

where the last inequality follows from the fact that $\|z\|_\infty \leq \|u\|_\infty$ since $z_k = \theta^k u_k$ and $|\theta| < 1$. This implies that $\Delta_{V_k} = \tilde{V}_k - V_k$ is such that

$$\max_k |(\Delta_{V_k})_{r,s}| \leq \mu n (\zeta + \gamma \log_2 n) \max_k |(U_k)_{r,s}|,$$

which yields

$$\|\Delta_{V_k}\|_\infty \leq m \max_k |(\Delta_{V_k})_{r,s}| \leq \mu m n (\zeta + \gamma \log_2 n) \max_{r,s,h} |(U_h)_{r,s}|. \quad (18)$$

Concerning the second part of the computation, for the matrix $\Delta_{W_k} = \widetilde{W}_k - W_k$ we have

$$\Delta_{W_k} = \text{fl}(e^{\tilde{V}_k}) - e^{V_k}, \quad \text{fl}(e^{\tilde{V}_k}) = e^{\tilde{V}_k} + E_k \quad (19)$$

where E_k is the error generated by computing the matrix exponential in floating point arithmetic. Here we assume that $\|E_k\|_\infty \leq \mu\tau\|W_k\|_\infty$ for some positive constant τ which depends on the algorithm used to compute the matrix exponential. From the properties of the Gâteaux derivative one has $\|e^{\tilde{V}_k} - e^{V_k}\| \doteq \|G(V_k, \Delta_{V_k})\|$, and from Theorem 2, applied with $x_i = \bar{\omega}_n^{ik} \theta^i$, $i = 1, \dots, n-1$, it follows that $\|G(V_k, \Delta_{V_k})\|_\infty \leq \|\Delta_{V_k}\|_\infty$ and $\|W_k\|_\infty \leq 1$.

Combining these results with (19) leads to the bound

$$\|\Delta_{W_k}\|_\infty \leq \|\Delta_{V_k}\|_\infty + \mu\tau. \quad (20)$$

Finally, for the third part of the computation, consisting of steps 4 and 5, we have

$$\delta_r \doteq \frac{1}{n} F \delta_w + \frac{1}{n} \delta''$$

where δ'' is the error obtained by computing Fw in floating point arithmetic. Thus from (16) we have

$$\begin{aligned}\|\delta_r\|_\infty &\leq \frac{1}{n}\|F\delta_w\|_\infty + \mu\gamma\log_2 n\|r\|_2 \\ &\leq \|\delta_w\|_\infty + \mu\gamma\sqrt{n}\log_2 n\|y\|_\infty,\end{aligned}\tag{21}$$

where the second inequality holds from (15) and from $r_k = \theta^k y_k$ since $|\theta| \leq 1$.

Moreover, we find that

$$(\delta_y)_k = \theta^{-k}(\delta_r)_k + \theta^{-k}r_k\nu_k = \theta^{-k}((\delta_r)_k + y_k\nu_k), \quad |\nu_k| \leq \zeta\mu. \tag{22}$$

Now we are ready to combine all the pieces and obtain the error bound on the computed value Y . From (22) we get

$$|(\delta_y)_k| \leq |\theta|^{-k}(\|\delta_r\|_\infty + \zeta\mu\|y\|_\infty).$$

On the other hand, by using (21), we find that

$$|(\delta_y)_k| \leq |\theta|^{-k}(\|\delta_w\|_\infty + (\zeta + \gamma\sqrt{n}\log_2 n)\mu\|y\|_\infty).$$

Thus we have

$$\|\Delta_{Y_k}\|_\infty \leq m|(\delta_y)_k| \leq m|\theta|^{-k}(\max_h \|\Delta_{W_h}\|_\infty + (\zeta + \gamma\sqrt{n}\log_2 n)\mu \max_h \|Y_h\|_\infty).$$

Moreover, from (20) and (18) we conclude with the following bound

$$\|\Delta_{Y_k}\|_\infty \leq m|\theta|^{-k}(\max_h \|\Delta_{V_h}\|_\infty + \mu\tau + (\zeta + \gamma\sqrt{n}\log_2 n)\mu \max_h \|Y_h\|_\infty).$$

Whence

$$\|\Delta_{Y_k}\|_\infty \leq \mu m|\theta|^{-k}(mn(\zeta + \gamma\log_2 n) \max_{r,s,h} |(U_h)_{r,s}| + \tau + (\zeta + \gamma\sqrt{n}\log_2 n) \max_h \|Y_h\|_\infty)$$

and we may conclude with the following

Theorem 6 *Let \hat{Y}_k be the value of Y_k provided by Algorithm 3 applied in floating point arithmetic with precision μ for computing $C_\epsilon(Y) = e^{C_\epsilon(U)}$, where $Y = (Y_k)_{k=0,n-1}$, $U = (U_k)_{k=0,n-1}$. Denote $\Delta_{Y_k} = Y_k - \hat{Y}_k$. One has*

$$\|\Delta_{Y_k}\|_\infty \leq \mu\epsilon^{-1}m\varphi$$

where

$$\varphi = mn(\zeta + \gamma\log_2 n) \max_{r,s,h} |(U_h)_{r,s}| + (\zeta + \gamma\sqrt{n}\log_2 n) \max_h \|Y_h\|_\infty + \tau,$$

$\zeta \doteq 1 + 2\sqrt{2}$, $\gamma \doteq 4\sqrt{2} + 1$, and $\tau\mu$ is the error bound in the computation of the matrix exponential, i.e., such that $\|fl(e^V) - e^V\|_\infty \leq \mu\tau\|e^V\|_\infty$ for an $m \times m$ matrix V .

In the case where $\epsilon = 1$, we apply Algorithm 4 to compute the exponential of a block-circulant matrix and the above result leads to

Theorem 7 *Let \hat{Y}_k be the value of Y_k provided by Algorithm 4 applied in floating point arithmetic with precision μ for computing $\mathcal{C}(Y) = e^{\mathcal{C}(U)}$, where $Y = (Y_k)_{k=0, n-1}$, $U = (U_k)_{k=0, n-1}$. Denote $\Delta_{Y_k} = Y_k - \hat{Y}_k$. One has*

$$\|\Delta_{Y_k}\|_\infty \leq \mu m \chi$$

where $\chi = mn\gamma \log_2 n \max_{r,s,h} |(U_h)_{r,s}| + \gamma\sqrt{n} \log_2 n \max_h \|Y_h\|_\infty + \tau$, and $\zeta \doteq 1 + 2\sqrt{2}$, $\gamma \doteq 4\sqrt{2} + 1$, and $\tau\mu$ is the error bound in the computation of the matrix exponential, i.e., such that $\|fl(e^V) - e^V\|_\infty \leq \mu\tau\|e^V\|_\infty$ for an $m \times m$ matrix V .

5 The exponential of a block-triangular block-Toeplitz matrix

Let $U = (U_i)_{i=0, n-1}$ be the block-vector defining the first block-row of the subgenerator $\mathcal{T}(U)$ of (1). Since block-triangular block-Toeplitz matrices form a matrix algebra, by using the Taylor series expansion of the matrix exponential, it follows that $e^{\mathcal{T}(U)}$ is still a block-triangular block-Toeplitz matrix. Denote by $A = (A_i)_{i=0, n-1}$ the block-vector defining the entries on the first block-row of $e^{\mathcal{T}(U)}$, i.e., such that $\mathcal{T}(A) = e^{\mathcal{T}(U)}$. In particular, we have $A_0 = e^{U_0}$.

Let $K \geq n$ and define the K dimensional block-vector $U^{(K)}$ obtained by completing U with zeros:

$$U^{(K)} = (U_i^{(K)})_{i=0, K-1}, \quad U_i^{(K)} = \begin{cases} U_i & \text{for } i = 0, \dots, n-1, \\ 0 & \text{for } i = n, \dots, K-1. \end{cases} \quad (23)$$

Consider the $K \times K$ block-triangular block-Toeplitz matrix $\mathcal{T}(U^{(K)})$. In view of [11, Theorem 3.6], if $K_2 > K_1 \geq n$, then $e^{\mathcal{T}(U^{(K_1)})}$ is the principal $K_1 \times K_1$ block-submatrix of $e^{\mathcal{T}(U^{(K_2)})}$. Denote by $A^{(K)} = (A_i)_{i=0, K-1}$ the block-vector defining the first block-row of $e^{\mathcal{T}(U^{(K)})}$, i.e., $A^{(K)}$ is the block-vector such that $\mathcal{T}(A^{(K)}) = e^{\mathcal{T}(U^{(K)})}$.

Let $\hat{U} = (\hat{U}_i)_{i=0, n-1}$ be such that

$$\hat{U}_0 = U_0 + \alpha I, \quad \hat{U}_i = U_i, \quad i = 1, \dots, n-1, \quad (24)$$

where $\alpha = \max_j (-(U_0)_{j,j})$. Define the block-vector $\hat{U}^{(K)}$ with block-components $\hat{U}_i^{(K)} = \hat{U}_i$ for $i = 0, \dots, n-1$, and $\hat{U}_i^{(K)} = 0$ for $i = n, \dots, K-1$. Observe that $\mathcal{T}(\hat{U}^{(K)}) = \mathcal{T}(U^{(K)}) + \alpha I$ is a nonnegative matrix, and we may write $e^{\mathcal{T}(U^{(K)})} = e^{-\alpha} e^{\mathcal{T}(\hat{U}^{(K)})}$. We denote by $\hat{A}^{(K)} = (\hat{A}_i)_{i=0, K-1}$ the block-vector such that $\mathcal{T}(\hat{A}^{(K)}) = e^{\mathcal{T}(\hat{U}^{(K)})}$. In particular we have $A_i = e^{-\alpha} \hat{A}_i$, $i = 0, \dots, K-1$.

5.1 Decay properties

In this section we investigate decay properties of the exponential $e^{\mathcal{T}(U^{(K)})}$ of a subgenerator, in the case where the subgenerator is a banded block-triangular block-Toeplitz matrix. These properties will be used in Section 5.3 to estimate the approximation error of the numerical method based on the embedding into a block-circulant matrix.

Decay properties of matrix functions have been analyzed in the literature. We refer the reader to the survey paper [5] and to [4]. In our case the structure and sign properties play an important role. The matrix exponential $e^{\mathcal{T}(U^{(K)})}$ is not banded in general, but its off-diagonal entries have useful decay properties for $K \rightarrow \infty$. To prove this fact we need the following result [8, Theorem 3.6] on decay properties of analytic functions.

Theorem 8 *Let $H(z) = \sum_{i=0}^{\infty} z^i H_i$ be an $m \times m$ matrix power series analytic for $z \in \mathbb{C}$ with $|z| < R$, and $R > 1$. For any $1 < \sigma < R$, the block-coefficients satisfy*

$$|H_i| \leq M(\sigma) \sigma^{-i}, \quad i = 0, 1, \dots \quad (25)$$

where $M(\sigma)$ is the $m \times m$ matrix with elements $\max_{|z|=\sigma} |h_{rs}(z)|$, for $r, s = 1, \dots, m$, and the inequality (25) is meant componentwise.

The following result provides bounds to A_i , $i = 0, \dots, K-1$.

Theorem 9 *Let $K \geq n$ and let $\mathcal{T}(A^{(K)}) = e^{\mathcal{T}(U^{(K)})}$, with $A^{(K)} = (A_i)_{i=0, K-1}$, where $\mathcal{T}(U)$ in (1) is a subgenerator and $U^{(K)}$ is defined in (23). For any $\sigma > 1$, we have*

$$A_i \mathbf{1} \leq e^{\alpha(\sigma^{n-1}-1)} \sigma^{-i} \mathbf{1}, \quad i = 0, \dots, K-1,$$

where $\alpha = \max_j (-(U_0)_{j,j})$.

Proof. We associate with the block-vector $\hat{U} = (\hat{U}_i)_{i=0, n-1}$ of (24) the $m \times m$ matrix polynomial $\hat{U}(z) = \sum_{h=0}^{n-1} z^h \hat{U}_h$. For the properties of block triangular block-Toeplitz matrices [8], the matrix $\mathcal{T}(\hat{U}^{(K)})^j$ is still a block-triangular block-Toeplitz matrix and the blocks in its first row are the coefficients of the matrix polynomial $P^{(j)}(z) = \hat{U}(z)^j \mod z^K$. Let $P_i^{(j)}$ be the matrix coefficient of degree i of $P^{(j)}(z)$, for $i = 0, \dots, K-1$. From the power series expression of the matrix exponential we find that

$$\hat{A}_i = \sum_{j=0}^{\infty} \frac{1}{j!} P_i^{(j)}, \quad i = 0, \dots, K-1. \quad (26)$$

We want to give an upper bound to the matrices $P_i^{(j)}$. Since $\hat{U}(z)^j$ is a matrix polynomial, then it is analytic in all the complex plane and we may apply Theorem 8 with $H(z) = \hat{U}(z)^j$ and any $\sigma > 1$. We have to estimate the matrix $M(\sigma)$. The matrix coefficients of $\hat{U}(z)$ are nonnegative, therefore for any $\sigma > 1$ and for any $z \in \mathbb{C}$ with $|z| = \sigma$, we have $|\hat{U}(z)^j| \leq \hat{U}(\sigma)^j \leq (\hat{U}(1)\sigma^{n-1})^j$. Since

$\mathcal{T}(U)$ is a subgenerator then $\hat{U}(1)\mathbf{1} = (\alpha I + \sum_{h=0}^{n-1} U_h)\mathbf{1} \leq \alpha\mathbf{1}$. So that we obtain $|\hat{U}(z)^j|\mathbf{1} \leq \alpha^j \sigma^{(n-1)j}\mathbf{1}$. Since $P^{(j)}(z) = \hat{U}(z)^j \pmod{z^K}$, then $P_i^{(j)}$ is the matrix coefficient of degree i of $\hat{U}(z)^j$ and, in view of (25), we find that $P_i^{(j)}\mathbf{1} \leq \alpha^j \sigma^{(n-1)j} \sigma^{-i}\mathbf{1}$. From this inequality and from (26) we obtain that for any $\sigma > 1$ and for $i = 0, \dots, K-1$

$$\hat{A}_i\mathbf{1} \leq \sum_{j=0}^{\infty} \frac{1}{j!} \alpha^j \sigma^{(n-1)j} \sigma^{-i}\mathbf{1} = \sigma^{-i} e^{\alpha\sigma^{n-1}}\mathbf{1}.$$

Since $\hat{A}_i = e^{-\alpha} A_i$ we conclude the proof. \square

5.2 Method based on ϵ -circulant matrix

Let $\epsilon \in \mathbb{C}$ with $|\epsilon|$ sufficiently small, and consider the block- ϵ -circulant matrix

$$\mathcal{C}_\epsilon(U) = \begin{bmatrix} U_0 & U_1 & \dots & U_{n-1} \\ \epsilon U_{n-1} & U_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & U_1 \\ \epsilon U_1 & \dots & \epsilon U_{n-1} & U_0 \end{bmatrix}. \quad (27)$$

The exponential of $\mathcal{C}_\epsilon(U)$ is still a block- ϵ -circulant matrix, that can be computed by means of Algorithm 3. Denote by $Y = (Y_i)_{i=0, n-1}$ the block-vector such that $\mathcal{C}_\epsilon(Y) = e^{\mathcal{C}_\epsilon(U)}$. The idea is to approximate the blocks A_i , defining $\mathcal{T}(A) = e^{\mathcal{T}(U)}$, by the matrices Y_i , for $i = 0, \dots, n-1$.

In order to estimate the approximation error, observe that the matrix $\mathcal{C}_\epsilon(U)$ can be written as $\mathcal{C}_\epsilon(U) = \mathcal{T}(U) + \epsilon L$, where

$$L = \begin{bmatrix} 0 & 0 & \dots & 0 \\ U_{n-1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ U_1 & \dots & U_{n-1} & 0 \end{bmatrix}. \quad (28)$$

This property allows to give the following estimate:

Theorem 10 *Assume that $\mathcal{T}(U)$ is a subgenerator, and that $\epsilon \in \mathbb{C}$. One has*

$$\|e^{\mathcal{T}(U)} - e^{\mathcal{C}_\epsilon(U)}\|_\infty \leq e^{|\epsilon|\|L\|_\infty} - 1.$$

Moreover, if ϵ is a pure imaginary number, then

$$\|e^{\mathcal{T}(U)} - \text{Re}(e^{\mathcal{C}_\epsilon(U)})\|_\infty \leq e^{|\epsilon|^2\|L\|_\infty^2} - 1,$$

where $\text{Re}(e^{\mathcal{C}_\epsilon(U)})$ is the real part of $e^{\mathcal{C}_\epsilon(U)}$.

Proof. According to (6),

$$e^{\mathcal{C}_\epsilon(U)} = e^{\mathcal{T}(U)} + \sum_{j=1}^{\infty} \frac{\epsilon^j}{j!} G^{[j]}(\mathcal{T}(U), L), \quad (29)$$

where $G^{[j]}(\mathcal{T}(U), L)$ are defined by means of (7). From Proposition 1 we obtain

$$\|e^{\mathcal{T}(U)} - e^{\mathcal{C}_\epsilon(U)}\|_\infty \leq \sum_{j=1}^{\infty} \frac{|\epsilon|^j}{j!} \|G^{[j]}(\mathcal{T}(U), L)\|_\infty \leq \sum_{j=1}^{\infty} \frac{|\epsilon|^j}{j!} \|L\|_\infty^j = e^{|\epsilon|\|L\|_\infty} - 1.$$

If ϵ is a pure imaginary number, since $e^{\mathcal{T}(U)}$ is a real matrix, the inequality is obtained by comparing the real parts in (29) and by applying Proposition 1. \square

It is interesting to observe that the choice of an imaginary value for ϵ provides an approximation error of the order $O(|\epsilon|^2)$ instead of $O(|\epsilon|)$. The idea of using an imaginary value for ϵ was used in [1] in the framework of Fréchet derivative approximation of matrix functions.

The error bound can be improved by performing the computation with several different values of ϵ and taking the mean of the real parts of the results obtained this way. For instance, choose $\epsilon_1 = (1 + \mathbf{i})\sqrt{2}\epsilon$, $\epsilon_2 = -\epsilon_1$, where $\epsilon > 0$, and recall that $e^{\mathcal{C}_{\epsilon_j}(U)}$, $j = 1, 2$ are power series in ϵ . Taking the arithmetic mean of $e^{\mathcal{C}_{\epsilon_1}(U)}$ and $e^{\mathcal{C}_{\epsilon_2}(U)}$, the components of odd degree in ϵ cancel out while the coefficient of ϵ^2 is pure imaginary. Therefore taking the real part of the arithmetic mean provides an error $O(\epsilon^4)$.

This technique can be generalized as follows. Choose an integer $k \geq 2$ and set $\epsilon_j = (\mathbf{i})^{1/k} \omega_k^j \epsilon$, $j = 0, \dots, k-1$, where $(\mathbf{i})^{1/k}$ is a principal k -th root of \mathbf{i} . Then one can verify that the arithmetic mean of $e^{\mathcal{C}_{\epsilon_j}(U)}$, $j = 0, \dots, k-1$ is a power series in ϵ^k , moreover, ϵ_j^k is a pure imaginary number so that the real part of this mean provides an approximation with error $O(\epsilon^{2k})$.

Observe that computing the exponential for different values of ϵ might seem a substantial computational overload. However, in a parallel model of computation, the exponentials $e^{\mathcal{C}_{\epsilon_j}(U)}$, $j = 0, \dots, k-1$, can be computed simultaneously by different processors at the same cost of computing a single exponential.

Algorithm 5 reports this averaging technique.

Theorem 10 provides us with a bound on the error generated by approximating the exponential of a block-upper triangular Toeplitz matrix by means of the exponential of a block- ϵ -circulant matrix. In fact, in practical computations in floating point arithmetic, the overall error is formed by two components: one component is given by the approximation error analyzed in Theorem 10, the second component is due to the roundoff and is estimated by Theorem 6. More precisely, the effectively computed approximation in floating point arithmetic is the block-vector with components $\tilde{Y}_k = Y_k + \Delta_{Y_k}$, $k = 0, \dots, n-1$, where $\|\Delta_{Y_k}\|_\infty$ is bounded in Theorem 6. On the other hand, $Y_k = A_k + E'_k$ where, by Theorem 10, E'_k is such that

$$\|[E'_0, \dots, E'_{n-1}]\|_\infty \leq \begin{cases} \psi(|\epsilon|^2 \|L\|_\infty^2) & \text{if } \epsilon \text{ is imaginary} \\ \psi(|\epsilon| \|L\|_\infty) & \text{otherwise} \end{cases}$$

Algorithm 5: Exponential of a block-triangular matrix by means of ϵ -circulant matrices and averaging

Input : The block-vector $U = (U_i)_{i=0,n-1}$ defining the first block-row of $\mathcal{T}(U)$; a real number $\epsilon > 0$; an integer $k > 0$.
Output : The block-vector $Y = (Y_i)_{i=0,n-1}$ that is an approximation of the first block-row of $e^{\mathcal{T}(U)}$.

- 1 Set $\epsilon_j = (\mathbf{i})^{1/k} \omega_k^j \epsilon$, $j = 0, \dots, k-1$
 - 2 Compute the first block row $W^{(j)}$ of $e^{\mathcal{C}_{\epsilon_j}(U)}$, $j = 0, \dots, k-1$, by means of Algorithm 3.
 - 3 Set $Y = \frac{1}{k} \sum_{j=0}^{k-1} \text{Re}(W^{(j)})$
-

where $\psi(t) = e^t - 1$. This way, for the overall error $E_k = \Delta_{Y_k} + E'_k$ one has

$$\|E_k\|_\infty \leq \|\Delta_{Y_k}\|_\infty + \|E'_k\|_\infty \leq m\mu\epsilon^{-1}\varphi + \psi(t),$$

for $t = |\epsilon|\|L\|_\infty$, or $t = |\epsilon|^2\|L\|_\infty^2$.

This shows the need to find a proper balance between the two errors: small values for $|\epsilon|$ provide a small approximation error $\|E'_k\|_\infty$ but the roundoff errors diverge to infinity as $\epsilon \rightarrow 0$. A good compromise is to choose ϵ so that the upper bounds to $\|E'_k\|_\infty$ and $\|\Delta_{Y_k}\|_\infty$ have the same order of magnitude. Equating these upper bounds in the case of non-imaginary ϵ yields

$$|\epsilon| = \sqrt{m\mu\varphi/\|L\|_\infty}, \quad \|E_k\|_\infty \leq 2\epsilon\|L\|_\infty$$

and in the case of imaginary ϵ ,

$$|\epsilon| = \sqrt[3]{m\mu\varphi/\|L\|_\infty^2}, \quad \|E_k\|_\infty \leq 2\epsilon^2\|L\|_\infty^2.$$

The latter bound is an $O(\mu^{2/3})$. This implies that asymptotically, as $\mu \rightarrow 0$, we may lose 1/3 of the digits provided by the floating point arithmetic.

If we adopt the strategy of performing the computation with k different values of $\epsilon_j = (\mathbf{i})^{1/k} \omega_k^j \epsilon$, $j = 0, \dots, k-1$, so that the approximation error is $O(\epsilon^{2k})$, then the total error turns to $O(\mu^{2k/2k+1})$, i.e., only $1/(2k+1)$ digits are lost.

An interesting point is that the quantities $\|L\|_\infty$ and $\max_{r,s,h} |(U_h)_{r,s}|$ are involved in the expressions of the error bound. Since $\mathcal{T}(U)$ is a generator, both these quantities are bounded from above by $\alpha = \max_j -(U_0)_{j,j}$. However, by means of simple manipulations, we may scale the input so that it is bounded by 1. This is performed by applying to $\mathcal{T}(U)$ the scaling and squaring technique of [12].

Let $p \geq 0$ be an integer such that $\alpha \leq 2^p$. Then, since $e^{\mathcal{T}(U)} = (e^{\mathcal{T}(U/2^p)})^{2^p}$, we first compute $e^{\mathcal{T}(U/2^p)}$ and then recover $e^{\mathcal{T}(U)}$ by performing p repeated matrix squaring. In this way we have $\|L/2^p\|_\infty < 1$ and $\max_{r,s,h} |(U_h)_{r,s}/2^p| < 1$. Since $\mathcal{T}(U/2^p)$ is still a generator, the error analysis performed for $e^{\mathcal{T}(U)}$

applies as well, and we can approximate the first block-row of $e^{\mathcal{T}(U/2^p)}$ with the first block-row $Y = (Y_i)$ of $e^{\mathcal{T}(U_\epsilon/2^p)}$ for a suitable $\epsilon \in \mathbb{C}$ with $|\epsilon| < 1$. Finally we recover an approximation to $e^{\mathcal{T}(U)}$ by computing $\mathcal{T}(Y)^{2^p}$ by means of p repeated squarings, by using the Toeplitz structure and Algorithm 2, in view of Remark 4. The overall procedure is described in Algorithm 6.

Algorithm 6: Exponential of a block-triangular block-Toeplitz matrix by using ϵ -circulant matrices

Input : The block-vector $U = (U_i)_{i=0,n-1}$ defining the first block-row of $\mathcal{T}(U)$, $\epsilon \in \mathbb{C}$
Output : The block-vector $Y = (Y_i)_{i=0,n-1}$, that is an approximation of the first block-row of $e^{\mathcal{T}(U)}$

- 1 $\alpha = \max_j (-(U_0)_{j,j})$, $p = \lfloor \log_2 \alpha \rfloor + 1$ and $\tilde{U} = U/2^p$
- 2 Compute Y , the first block-row of $e^{\mathcal{C}_\epsilon(\tilde{U})}$, by means of Algorithm 3
- 3 If ϵ is imaginary, replace Y with the real part of Y
- 4 **for** $r = 1, \dots, p$ **do**
- 5 | compute $\mathcal{T}(Y) = \mathcal{T}(Y)\mathcal{T}(Y)$
- 6 **end for**

5.3 Embedding into a circulant matrix

The idea of this method is to embed the matrix $\mathcal{T}(U)$ into a $K \times K$ block-circulant matrix $\mathcal{C}(U^{(K)})$. The first block-row of $e^{\mathcal{T}(U)}$ is approximated by the first n blocks of the first block-row of $e^{\mathcal{C}(U^{(K)})}$. Specifically, take $K \geq n$ and consider the block-vector $U^{(K)}$ defined in (23). The block-circulant matrix $\mathcal{C}(U^{(K)})$ may be partitioned as

$$\mathcal{C}(U^{(K)}) = \begin{bmatrix} \mathcal{T}(U) & P \\ Q & \mathcal{T}(U^{(K-n)}) \end{bmatrix},$$

where P and Q are $n \times (K-n)$ and $(K-n) \times n$ block-matrices, respectively.

Denote by \mathcal{E}_1 and by \mathcal{E}_K the $(mnK) \times (mn)$ matrices formed by the first mn and the last mn columns, respectively, of the identity matrix of size mnK . The matrix $\mathcal{C}(U^{(K)})$ can be also written as

$$\mathcal{C}(U^{(K)}) = \mathcal{T}(U^{(K)}) + H_K, \quad H_K = \mathcal{E}_K L \mathcal{E}_1^T, \quad (30)$$

where the matrix L is defined in (28). Because of the triangular Toeplitz structure, the desired matrix $e^{\mathcal{T}(U)}$ is identical to the $n \times n$ block-leading submatrix of $e^{\mathcal{T}_K(U)}$. Our idea is to approximate the first block-row of $e^{\mathcal{T}(U)}$ with the first n blocks of the first row of $e^{\mathcal{C}(U^{(K)})}$. As pointed out in Section 4, $e^{\mathcal{C}(U^{(K)})}$ is a block-circulant matrix, and can be computed by means of Algorithm 4 with $3m^2K \log_2 K + m^2K$ ops, plus the cost of computing K exponentials of $m \times m$ matrices.

Denote by $S^{(K)} = (S_i^{(K)})_{i=0, K-1}$ the first block-row of $e^{\mathcal{C}(U^{(K)})}$, so that $\mathcal{C}(S^{(K)}) = e^{\mathcal{C}(U^{(K)})}$. An approximation of the matrices A_i , $i = 0, \dots, n-1$, defining the first block-row of $e^{\mathcal{T}(U)}$ is provided by $S_i^{(K)}$, $i = 0, \dots, n-1$; as K increases, the approximation improves, as shown by the following result.

Theorem 11 *Let $e^{\mathcal{T}(U)} = \mathcal{T}(A)$, with $A = (A_i)_{i=0, n-1}$. Let $K \geq n$ and let $e^{\mathcal{C}(U^{(K)})} = \mathcal{C}(S^{(K)})$, with $S^{(K)} = (S_i^{(K)})_{i=0, K-1}$. One has $S_i^{(K)} - A_i \geq 0$ for $i = 0, \dots, n-1$, and*

$$\left\| \begin{bmatrix} S_0^{(K)} - A_0 & \dots & S_{n-1}^{(K)} - A_{n-1} \end{bmatrix} \right\|_{\infty} \leq f_K(\sigma) \quad (31)$$

for any $\sigma > 1$, where

$$f_K(\sigma) = (e^{\|L\|_{\infty}} - 1)e^{\alpha(\sigma^{n-1}-1)} \frac{\sigma^{-K+n}}{1 - \sigma^{-1}},$$

with $\alpha = \max_j (-(U_0)_{j,j})$ and L defined in (28).

Proof. By using (30) and (6), we find that

$$e^{\mathcal{C}(U^{(K)})} - e^{\mathcal{T}(U^{(K)})} = \sum_{j=1}^{\infty} \frac{1}{j!} G^{[j]}(\mathcal{T}(U^{(K)}), H_K).$$

Equating the first n blocks in the first block-row in the above equation yields

$$\begin{bmatrix} S_0^{(K)} - A_0 & \dots & S_{n-1}^{(K)} - A_{n-1} \end{bmatrix} = \sum_{j=1}^{\infty} \frac{1}{j!} W^{[j]}, \quad (32)$$

where $W^{[j]}$ is the block-row vector formed by the first n block-entries in the first block-row of $G^{[j]}(\mathcal{T}_K(U), H_K)$. That is,

$$W^{[j]} = \widehat{\mathcal{E}}_1^T G^{[j]}(\mathcal{T}(U^{(K)}), H_K) \mathcal{E}_1,$$

where $\widehat{\mathcal{E}}_1$ is the $mnK \times m$ matrix formed by the first m columns of the identity matrix. Since $H_K \geq 0$, from (32) and from Proposition 1 we deduce that $W^{[j]} \geq 0$ so that $S_i^{(K)} - A_i \geq 0$ for $i = 0, \dots, n-1$. On the other hand, in view of (7) and from the fact that $H_K = \mathcal{E}_K L \mathcal{E}_1^T$, we may write

$$\begin{aligned} W^{[j]} &= j \int_0^1 \widehat{\mathcal{E}}_1^T e^{(1-s)\mathcal{T}(U^{(K)})} \mathcal{E}_K L \mathcal{E}_1^T G^{[j-1]}(s\mathcal{T}(U^{(K)}), H_K) \mathcal{E}_1 ds \\ &= j \int_0^1 V(s) L Z^{[j-1]}(s) ds \end{aligned} \quad (33)$$

where $V(s) = \begin{bmatrix} V_0(s) & \dots & V_{n-1}(s) \end{bmatrix}$ is the block-row vector formed by the last n block-entries of the first block-row of $e^{(1-s)\mathcal{T}(U^{(K)})}$, and $Z^{[j-1]}(s)$ is the $n \times n$ block leading submatrix of $G^{[j-1]}(s\mathcal{T}(U^{(K)}), H_K)$.

Since the matrix $(1-s)\mathcal{T}(U^{(K)})$ is a subgenerator, it follows that $V_i(s) \geq 0$ and, from Theorem 9, that for any $\sigma > 1$,

$$V_i(s)\mathbf{1} \leq e^{(1-s)\alpha(\sigma^{n-1}-1)}\sigma^{-K+n-i}\mathbf{1} \leq e^{\alpha(\sigma^{n-1}-1)}\sigma^{-K+n-i}\mathbf{1},$$

for $i = 0, \dots, n-1$, where the latter inequality follows from the fact that $1-s \leq 1$. This implies that

$$\|V(s)\|_\infty \leq e^{\alpha(\sigma^{n-1}-1)} \sum_{i=0}^{n-1} \sigma^{-K+n-i} \leq e^{\alpha(\sigma^{n-1}-1)} \frac{\sigma^{-K+n}}{1-\sigma^{-1}}.$$

Moreover, since $s\mathcal{T}(U^{(K)})$ is a subgenerator, H_K is nonnegative and $\|H_K\|_\infty = \|L\|_\infty$, then, from Proposition 1, we have $G^{[j-1]}(s\mathcal{T}(U^{(K)}), H_K) \geq 0$ and

$$\|G^{[j-1]}(s\mathcal{T}(U^{(K)}), H_K)\|_\infty \leq s^{j-1}\|L\|_\infty^{j-1}.$$

This latter inequality implies that $\|Z^{[j-1]}(s)\|_\infty \leq s^{j-1}\|L\|_\infty^{j-1}$. Therefore, by taking norms in (33), we find that

$$\begin{aligned} \|W^{[j]}\|_\infty &\leq j \int_0^1 \|V(s)\|_\infty \|L\|_\infty \|Z^{[j-1]}(s)\|_\infty ds \\ &\leq j \|L\|_\infty^j e^{\alpha(\sigma^{n-1}-1)} \frac{\sigma^{-K+n}}{1-\sigma^{-1}} \int_0^1 s^{j-1} ds = \|L\|_\infty^j e^{\alpha(\sigma^{n-1}-1)} \frac{\sigma^{-K+n}}{1-\sigma^{-1}}. \end{aligned}$$

Hence, by taking norms in (32), we obtain (31). \square

Remark 12 The matrices A_i and $S_i^{(K)}$ have a probabilistic interpretation. Namely, the matrix A_i is the probability that the BMAP is absorbed after time 1, and at time 1 there have been $i < n$ arrivals; the matrix $S_i^{(K)}$ is the probability that the BMAP is absorbed after time 1, and at time 1 there have been i , or $i+K$, or $i+2K$, or \dots , arrivals. Clearly, there are more trajectories favourable for $S_i^{(K)}$ than for A_i and $A_i \leq S_i^{(K)}$. Similarly, there are more trajectories favourable for $S_i^{(K)}$ than for $S_i^{(\ell K)}$ for a positive integer ℓ . This shows that, if we take a sequence of integers ℓ_1, ℓ_2, \dots , and a sequence K_0, K_1, K_2, \dots , such that $K_{n+1} = \ell_{n+1}K_n$, then

$$S_i^{(K_0)} \geq S_i^{(K_1)} \geq S_i^{(K_2)} \geq \dots \geq A_i$$

for $i = 0, \dots, K_0 - 1$. Therefore, the sequence $\{S^{(K)}\}$ has some monotonicity property in its convergence to A .

The bound in (31) shows that the error has an exponential decay as K increases. Moreover, such bound holds for any $\sigma > 1$. Therefore we can fix a tolerance ϵ and a $\sigma > 1$, and find K such that $f_K(\sigma) < \epsilon$. Since we would like to keep K as low as possible, another way to proceed is to fix a tolerance ϵ and find σ such that the size K for which $f_K(\sigma) < \epsilon$ is minimum. More specifically, after

some manipulations, from the condition $f_K(\sigma) < \epsilon$ we obtain that $K > g(\sigma)$ where

$$g(\sigma) = \frac{\alpha(\sigma^{n-1} - 1) + \log(\sigma/(\sigma - 1)) + \log(\epsilon^{-1}) + \log(e^{\|L\|_\infty} - 1)}{\log(\sigma)} + n.$$

Since $\sigma > 1$ is arbitrary, we choose σ such that $g(\sigma)$ has a minimum value. In fact, the function $g(\sigma)$ diverges to infinity as σ tends to 1 and to ∞ , therefore it has at least a local minimum σ^* and we can choose $K > g(\sigma^*)$.

When we perform the computation in floating point arithmetic, we have to consider also the error generated by roundoff in computing the exponential of a block-circulant matrix. In practical computations, we obtain a block-vector with components $\widehat{Y}_i = Y_i + \Delta_{Y_i}$, $k = 0, \dots, n-1$, where $\|\Delta_{Y_i}\|_\infty$ is bounded in Theorem 7 and $Y_i = A_i + E'_i$ where, by Theorem 11, E'_i is such that

$$\|[E'_0, \dots, E'_{n-1}]\|_\infty \leq f_K(\sigma).$$

Altogether, for the overall error $E_i = \Delta_{Y_i} + E'_i$, one has

$$\|E_i\|_\infty \leq \|\Delta_{Y_i}\|_\infty + \|E'_i\|_\infty \leq m\mu\chi + f_K(\sigma).$$

A similar analysis can be carried out for the relative error. In this case the inequality $f_K(\sigma) < \epsilon$ is replaced by $f_K(\sigma) < \hat{\epsilon}$, for $\hat{\epsilon} = \epsilon\|[A_0, \dots, A_{n-1}]\|_\infty$. So that the function $g(\sigma)$ is modified by replacing ϵ with $\hat{\epsilon}$.

Like at the end of Section 5.2, in the overall estimate of the error, the quantities $\|L\|_\infty$ and $\max_{r,s,h} |(U_h)_{r,s}|$ are bounded from above by $\alpha = \max_j (-(U_0)_{j,j})$, and we may scale the block-vector U so that these quantities are bounded by 1.

The overall procedure is summarized in Algorithm 7, where the repeated squaring of the block-triangular block-Toeplitz matrices can be performed by using Algorithm 2, as explained in Remark 4.

Algorithm 7: Exponential of a block-triangular block-Toeplitz matrix by using embedding into a circulant matrix

Input : The block-vector $U = (U_i)_{i=0,n-1}$, an integer $K > n$
Output : The block-vector $Y = (Y_i)_{i=0,n-1}$, that is an approximation of the first block-row of $e^{\mathcal{T}(U)}$

- 1 Set $\alpha = \max_j (-(U_0)_{j,j})$, $p = \lfloor \log_2 \alpha \rfloor + 1$ and $\tilde{U} = U/2^p$
 - 2 Set $W = (W_i)_{i=0,K-1}$ with $W_i = \tilde{U}_i$ for $i = 0, \dots, n-1$, $W_i = 0$ for $i = n, \dots, K-1$
 - 3 Apply Algorithm 4 to compute the first block-row $V = (V_i)_{i=0,K-1}$ of $e^{\mathcal{C}(W)}$
 - 4 Set $Y_i = V_i$, for $i = 0, \dots, n-1$.
 - 5 **for** $r = 1, \dots, p$ **do**
 - 6 | compute $\mathcal{T}(Y) = \mathcal{T}(Y)\mathcal{T}(Y)$
 - 7 **end for**
-

5.4 Taylor series method

In this section we use the Taylor series method for computing the exponential of an essentially nonnegative matrix, where the block-triangular block-Toeplitz structure is exploited to perform fast matrix-vector multiplications. The computation of the exponential of an essentially nonnegative matrix have been analyzed in [20] and [17].

Following [20] and [17], the Taylor series method is applied to compute $e^{\mathcal{T}(\hat{U})}$, since the matrix $\mathcal{T}(\hat{U}) = \mathcal{T}(U) + \alpha I$ is nonnegative and $e^{\mathcal{T}(U)}$ can be obtained by means of the equation $e^{\mathcal{T}(U)} = e^{-\alpha} e^{\mathcal{T}(\hat{U})}$. In this way, we avoid possible cancellations in the Taylor summation.

Denote by $S_r(\mathcal{T}(\hat{U}))$ the Taylor series truncated at the r th term, namely

$$S_r(\mathcal{T}(\hat{U})) = \sum_{k=0}^{r-1} \frac{\mathcal{T}(\hat{U})^k}{k!}.$$

The following bound on the approximation error is given in [20].

Theorem 13 *Let r be such that $\rho(\mathcal{T}(\hat{U})/(r+1)) < 1$. Then*

$$|e^{\mathcal{T}(\hat{U})} - S_r(\mathcal{T}(\hat{U}))| \leq \frac{\mathcal{T}(\hat{U})^r}{r!} \left(I - \frac{\mathcal{T}(\hat{U})}{r+1} \right)^{-1}.$$

The scaling and squaring method is used to accelerate the convergence of the Taylor series, by using the property that

$$e^{\mathcal{T}(U)} = e^{-\alpha} e^{\mathcal{T}(\hat{U})} = \left(e^{-\alpha/2^p} e^{\mathcal{T}(\hat{U})/2^p} \right)^{2^p}.$$

Indeed, if $\tilde{\rho}$ is an estimate of $\rho(\mathcal{T}(\hat{U}))$, and if $p = \lfloor \log_2 \tilde{\rho} \rfloor + 1$, then $\rho(\mathcal{T}(\hat{U})/2^p) < 1$ and the truncated Taylor series expansion is used to approximate $e^{\mathcal{T}(\hat{U})/2^p}$. Since $\mathcal{T}(\hat{U})$ is block-triangular block-Toeplitz, then $\rho(\mathcal{T}(\hat{U})) = \rho(\hat{U}_0)$.

The Toeplitz structure is used in the computation of the Taylor expansion and in the squaring procedure. In fact, the computation of each term in the power series expansion consists in performing products between block-triangular block-Toeplitz matrices, that can be done by applying Algorithm 2 in view of Remark 4; similarly in the squaring procedure at the end of the algorithm.

Concerning rounding errors, we observe that the Taylor polynomial is the sum of nonnegative terms. Therefore no cancellation error is encountered in this summation. The main source of rounding errors is the computation of the powers $\mathcal{T}(\hat{U})^k$ for $k = 2, \dots$, which are computed by means of Algorithm 2 in view of Remark 4 relying on FFT. We omit the error analysis of this computation, which is standard. However, we recall that in view of Theorem 5, FFT is normwise backward stable but not component-wise stable. For this reason, for the truncation of the power series it is convenient to replace the

component-wise bound expressed by Theorem 13 by the norm-wise bound

$$\|e^{\mathcal{T}(\hat{U})} - S_r(\mathcal{T}(\hat{U}))\|_\infty \leq \left\| \frac{\mathcal{T}(\hat{U})^r}{r!} \left(I - \frac{\mathcal{T}(\hat{U})}{r+1} \right)^{-1} \right\|_\infty,$$

from which we obtain that the condition $\left\| \frac{\mathcal{T}(\hat{U})^r}{r!} \left(I - \frac{\mathcal{T}(\hat{U})}{r+1} \right)^{-1} \right\|_\infty < \epsilon \|S_r(\mathcal{T}(\hat{U}))\|_\infty$

implies that $\|e^{\mathcal{T}(\hat{U})} - S_r(\mathcal{T}(\hat{U}))\|_\infty \leq \epsilon \|S_r(\mathcal{T}(\hat{U}))\|_\infty$.

The overall procedure is stated in Algorithm 8.

It is worth pointing out that, if the computation of the powers of the triangular Toeplitz matrices is performed with the standard algorithm then the computation is component-wise stable as shown in [20].

Algorithm 8: Exponential of a block-triangular block-Toeplitz matrix by using Taylor series expansion

Input : The block-vector $U = (U_i)_{i=0,n-1}$ defining the first block-row of $\mathcal{T}(U)$, a tolerance $\epsilon > 0$, a maximum number of iterations K

Output : The block-vector $Y = (Y_i)_{i=0,n-1}$, that is an approximation of the first block-row of $e^{\mathcal{T}(U)}$

- 1 Set $\alpha = \max_j (-(U_0)_{j,j})$
- 2 Set $\hat{U} = (U_i)_{i=0,n-1}$, $\hat{U}_0 = U_0 + \alpha I$, $\hat{U}_i = U_i$, $i = 1, \dots, n-1$
- 3 Compute $\tilde{\rho}$ an estimate of $\rho(\hat{U}_0)$, or set $\tilde{\rho} = \|\hat{U}_0\|_\infty$
- 4 Compute $p = \lfloor \log_2 \tilde{\rho} \rfloor + 1$ and $V = \hat{U}/2^p$
- 5 Set $W = V$ and $Y = (Y_i)_{i=0,n-1}$, $Y_0 = I + V_0$, $Y_i = V_i$, $i = 1, \dots, n-1$.
- 6 **for** $r = 2, \dots, K$ **do**
- 7 Compute $\mathcal{T}(W) = \mathcal{T}(V)\mathcal{T}(W/r)$ and $Y = Y + W$
- 8 **if** $\|W\|_\infty < \epsilon \|Y\|_\infty$ **then**
- 9 **break**
- 10 **end if**
- 11 **end for**
- 12 Compute $Y = e^{-\alpha/2^p} Y$
- 13 **for** $i = 1, \dots, p$ **do**
- 14 compute $\mathcal{T}(Y) = \mathcal{T}(Y)\mathcal{T}(Y)$
- 15 **end for**

6 Numerical experiments

The numerical experiments have been performed in Matlab. To compute the error obtained with the proposed algorithms we have first computed the exponential by using the `vpa` arithmetic of the Symbolic Toolbox with 40 digits and we have considered this approximation as the exact value.

n	cw-abs	cw-rel	nw-abs	nw-rel
128	8.4e-14	5.8e-11	6.5e-12	8.0e-12
256	1.1e-14	8.9e-11	1.7e-12	2.1e-12
512	1.2e-14	2.6e-09	7.7e-13	9.6e-13
1024	2.2e-14	9.1e-04	1.9e-12	2.4e-12

Table 1: Errors generated by Algorithm 6, based on the ϵ -circulant technique, with $\epsilon = \mathbf{i} \cdot 10^{-2}$

Denote by \tilde{A}_h , $h = 0, \dots, n-1$, the approximations of the blocks on the first row of $e^{\mathcal{T}(U)}$ and define the four errors

$$\begin{aligned}
\text{cw-abs} &= \max_{h,i,j} |(A_h)_{i,j} - (\tilde{A}_h)_{i,j}|, \\
\text{cw-rel} &= \max_{h,i,j} \{ |(A_h)_{i,j} - (\tilde{A}_h)_{i,j}| / |(A_h)_{i,j}| \}, \\
\text{nw-abs} &= \|[A_0 - \tilde{A}_0, \dots, A_{n-1} - \tilde{A}_{n-1}]\|_\infty, \\
\text{nw-rel} &= \|[A_0 - \tilde{A}_0, \dots, A_{n-1} - \tilde{A}_{n-1}]\|_\infty / \|[A_0, \dots, A_{n-1}]\|_\infty,
\end{aligned}$$

which represent absolute/relative component-wise and norm-wise errors, respectively.

We compare the accuracy and the execution times of the proposed algorithms.

The test matrix $\mathcal{T}(U)$ is taken from two real world problems concerning the Erlangian approximation of a Markovian fluid queue [10]. The block-size n of $\mathcal{T}(U)$ is usually very large since a bigger n leads to a better Erlangian approximation, while the size m of the blocks is equal to 2 for both problems.

We show the performances in terms of accuracy of the algorithm based on the ϵ -circulant matrix. In Table 1 we report the errors generated by Algorithm 6 with $\epsilon = \mathbf{i} \cdot 10^{-2}$ applied to the first problem. Observe that the errors are much smaller in magnitude than $|\epsilon|$. The component-wise and norm-wise absolute errors range around $10^{-14} - 10^{-12}$, while the componentwise relative errors deteriorate as n increases; the norm-wise relative errors moderately increase as n increases. This behavior is expected since the use of FFT makes the algorithm stable in norm, while the component-wise accuracy is not guaranteed.

In Figure 1 we report the absolute/relative component-wise and the relative norm-wise errors as a function of $\epsilon = \mathbf{i} \cdot \theta$, with θ varying from 10^{-10} to 10^0 , in the case $n = 512$. In Figure 1a the scaling technique is not applied, while in Figure 1b the scaling is applied, as described in Algorithm 6. It is worth pointing out how the scaling allows to obtain a better accuracy, and the best performances are obtained with a larger value of $|\epsilon|$. Observe also that with the scaling technique the component-wise relative error takes values close to 10^{-9} while the theoretical bound is asymptotically $(2/3)\mu \approx 1.e - 10$. Another interesting remark is that the absolute component-wise errors and the relative norm-wise errors reach a minimum value for a moderately large value of θ , and

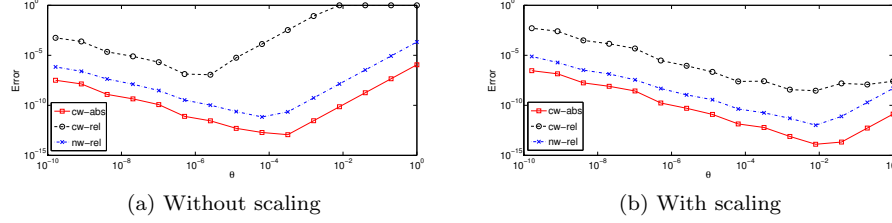


Figure 1: Error as function of $\epsilon = i\theta$ for the ϵ -circulant algorithm, with $n = 512$

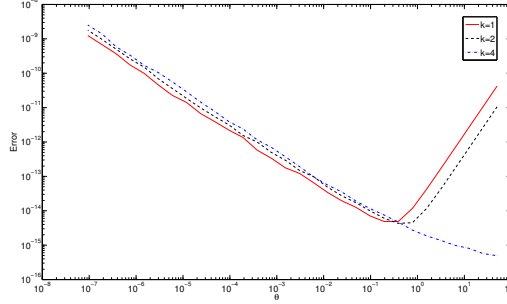


Figure 2: Error as function of θ for the ϵ -circulant algorithm, with k interpolation points and $n = 512$

substantially increase for values smaller than this minimum. This is due to the effect of round-off errors, which increase as $|\epsilon|$ goes to zero.

In Figure 2 we report the normwise relative errors obtained with the ϵ -circulant technique, described in Algorithm 5, applied with k different values $\epsilon_j = (i)^{1/k} \omega_k^j \theta$, for $j = 0, \dots, k-1$, where the solution is the arithmetic mean of $e^{C_{\epsilon_j}(U)}$. It is interesting to observe that using $k = 2$ leads to an approximation error better than $k = 1$, while for $k = 4$ the solution provided by the algorithm has an error close to the machine precision. Actually from this picture it is possible to figure out where the approximation errors and the roundoff errors dominate. For $k = 4$ the graph of the overall error is almost decreasing, this shows that the approximation error is removed by the technique of averaging the approximations obtained with different values of ϵ_j . From this behaviour one deduces that the approximation error numerically behaves like a polynomial of degree less than 8. This guess should be worth being investigated from a theoretical point of view.

Now consider the method based on the embedding into a circulant matrix. In Table 2 we report the errors generated by Algorithm 7 with $K = 4n$ applied to the first problem. The component-wise absolute errors are of the order of

n	cw-abs	cw-rel	nw-abs	nw-rel
128	2.0e-16	1.4e-12	8.9e-15	1.1e-14
256	4.0e-16	2.4e-11	2.2e-14	2.8e-14
512	8.5e-16	2.5e-09	4.3e-14	5.4e-14
1024	6.3e-16	3.4e-04	8.4e-14	1.0e-13

Table 2: Errors generated by Algorithm 7, based on the embedding technique, with $K = 4n$

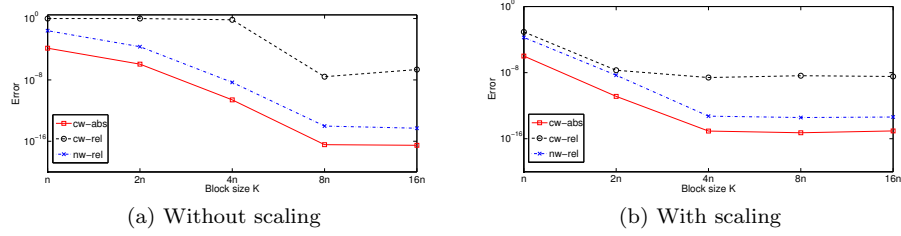


Figure 3: Error as function of K for the embedding algorithm, with $n = 512$

the machine precision, while the component-wise relative errors deteriorate as n increases; the norm-wise relative errors remain quite small as n increases. As for the ϵ -circulant method, this behavior is expected for the use of FFT. The accuracy of this algorithm is better than that obtained with the ϵ -circulant method.

In Figure 3 we report the absolute/relative component-wise and relative norm-wise errors as a function of K , in the case $n = 512$. In Figure 3a the scaling technique is not applied, while in Figure 3b the scaling is applied, as described in Algorithm 7. Also in this case it is worth pointing out how the scaling allows to obtain a better accuracy and optimal performances with smaller value of the block-size K , that is $4n$ vs. $8n$.

In Table 3 we report the errors generated by Algorithm 8 based on Taylor expansion. The errors have the same magnitude as those of Table 2 for the method based on the embedding.

In Table 4 we report the CPU time in seconds, as a function of n , needed by the algorithm based on ϵ -circulant matrix (**epc**), on embedding into a circulant

n	cw-abs	cw-rel	nw-abs	nw-rel
128	4.7e-16	9.5e-13	5.7e-15	7.0e-15
256	1.8e-15	4.3e-12	2.2e-14	2.8e-14
512	8.9e-16	1.3e-09	3.0e-14	3.8e-14
1024	4.8e-15	7.7e-04	1.3e-13	1.6e-13

Table 3: Errors generated by Algorithm 8, based on Taylor expansion

Algorithm \ n	256	512	1024	2048	4096
<code>epc</code>	0.2	0.5	1.5	4.6	16.0
<code>emb</code>	0.4	0.9	2.4	6.8	22.4
<code>taylor</code>	0.6	1.4	3.8	11.5	37.6
<code>expm</code>	0.9	5.9	327.7	*	*

Table 4: CPU time as function of the block-size n

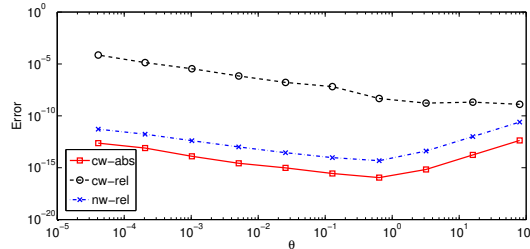


Figure 4: Error as function of $\epsilon = i\theta$ for the ϵ -circulant algorithm, with $n = 512$

matrix (`emb`), on Taylor series expansion (`taylor`) and by the `expm` function of Matlab. The symbol “*” denotes an execution time greater than 100 seconds. The time needed by `expm` increases much faster than the time needed by the other methods. The method `epc` is the fastest, and the method based on embedding is slightly faster than the Taylor series method when n is large enough.

Concerning the second problem, we report only the results in the case where scaling is applied. In fact, there is not much differences between the scaled and the unscaled versions since this problem is already well scaled in its original formulation. In Figure 4 we report the errors for the method based on ϵ -circulant matrices. It is interesting to note that the optimal value of $|\epsilon|$ is close to 1 and that the component-wise relative error is minimized by values of $|\epsilon|$ greater than 1. This fact, which apparently seems to be a contradiction, is explained as follows. Large values of ϵ generate large errors in the lower triangular part, i.e., the lower triangular part of $e^{\mathcal{T}(U)} - e^{\mathcal{C}_\epsilon(U)}$ has large norm. On the other hand we consider the first block-row of $e^{\mathcal{C}_\epsilon(U)}$ to approximate the matrix exponential of $\mathcal{T}(U)$, therefore the errors are not influenced by a large error in the lower triangular part.

In Figure 5 we report the errors for the algorithm based on embedding. It is relevant to observe that the errors are essentially minimized with an embedding of just double size.

To conclude, the method based on ϵ -circulant is the fastest one, but the accuracy of the results is lower than that provided by the embedding and Taylor series expansion. However, by applying the averaging technique we can dramatically improve the accuracy of the ϵ -circulant algorithm.

The computational time of all the structured algorithms is much lower than

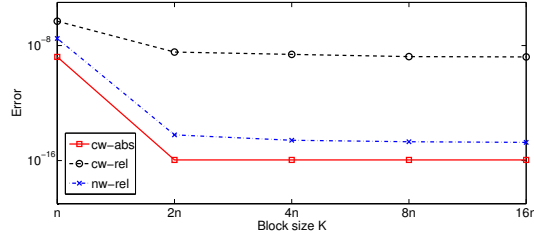


Figure 5: Error as function of K for the embedding algorithm, with $n = 512$

the cost of the general method implemented in the `expm` function of Matlab and allows to deal with matrices with huge size.

The algorithms based on embedding, on ϵ -circulant matrices are faster than the one based on Taylor series with FFT matrix arithmetic. Moreover they are better suited for a parallel implementation.

References

- [1] A. H. Al-Mohy, N. J. Higham, The complex step approximation to the Fréchet derivative of a matrix function, *Numer. Algorithms* 53 (1) (2010) 113–148. doi:10.1007/s11075-009-9323-y.
URL <http://dx.doi.org/10.1007/s11075-009-9323-y>
- [2] S. Asmussen, F. Avram, M. Usábel, Erlangian approximations for finite-horizon ruin probabilities, *ASTIN Bulletin* 32 (2002) 267–281.
- [3] N. Bean, M. O’Reilly, P. Taylor, Algorithms for return probabilities for stochastic fluid flows, *Stochastic Models* 21 (2005) 149–184.
- [4] M. Benzi, P. Boito, Decay properties for functions of matrices over C^* -algebras, *Linear Algebra Appl.* 456 (2014) 174–198. doi:10.1016/j.laa.2013.11.027.
URL <http://dx.doi.org/10.1016/j.laa.2013.11.027>
- [5] M. Benzi, P. Boito, N. Razouk, Decay properties of spectral projectors with applications to electronic structure, *SIAM Rev.* 55 (1) (2013) 3–64. doi:10.1137/100814019.
URL <http://dx.doi.org/10.1137/100814019>
- [6] D. Bini, Parallel solution of certain Toeplitz linear systems, *SIAM J. Comput.* 13 (2) (1984) 268–276. doi:10.1137/0213019.
URL <http://dx.doi.org/10.1137/0213019>
- [7] D. A. Bini, B. Iannazzo, B. Meini, Numerical Solution of Algebraic Riccati Equations, no. 9 in *Fundamentals of Algorithms*, SIAM, Philadelphia PA, 2012.

- [8] D. A. Bini, G. Latouche, B. Meini, Numerical methods for structured Markov chains, Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005, oxford Science Publications. doi:10.1093/acprof:oso/9780198527688.001.0001. URL <http://dx.doi.org/10.1093/acprof:oso/9780198527688.001.0001>
- [9] D. Bini, V. Pan, Polynomial and Matrix Computations, Birkhäuser, Boston, 1994.
- [10] S. Dendievel, G. Latouche, Approximation for time-dependent distributions in Markovian fluid models, SubmittedArXiv:1409.4989.
- [11] N. J. Higham, Functions of Matrices: Theory and Computation, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
- [12] N. J. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM Rev. 51 (4) (2009) 747–764. doi:10.1137/090768539. URL <http://dx.doi.org/10.1137/090768539>
- [13] S. Lee, H.-K. Pang, H.-W. Sun, Shift-invert Arnoldi approximation to the Toeplitz matrix exponential, SIAM Journal on Scientific Computing 32 (2) (2010) 774–792.
- [14] I. Najfeld, T. F. Havel, Derivatives of the matrix exponential and their computation, Adv. in Appl. Math. 16 (3) (1995) 321–375. doi:10.1006/aama.1995.1017. URL <http://dx.doi.org/10.1006/aama.1995.1017>
- [15] H.-K. Pang, H.-W. Sun, Shift-invert Lanczos method for the symmetric positive semidefinite Toeplitz matrix exponential, Numerical Linear Algebra with Applications 18 (3) (2011) 603–614.
- [16] V. Ramaswami, D. G. Woolford, D. A. Stanford, The Erlangization method for Markovian fluid flows, Ann. Oper. Res. 160 (2008) 215–225.
- [17] M. Shao, W. Gao, J. Xue, Aggressively truncated Taylor series method for accurate computation of exponentials of essentially nonnegative matrices, SIAM J. Matrix Anal. Appl. 35 (2) (2014) 317–338. doi:10.1137/120894294. URL <http://dx.doi.org/10.1137/120894294>
- [18] D. Stanford, F. Avram, A. Badescu, L. Breuer, A. da Silva Soares, G. Latouche, Phase-type approximations to finite-time ruin probabilities in the Sparre Andersen and stationary renewal risk models, ASTIN Bulletin 35 (2005) 131–144.
- [19] D. A. Stanford, G. Latouche, D. G. Woolford, D. Boychuk, A. Hunchak, Erlangized fluid queues with application to uncontrolled fire perimeter, Stochastic Models 21 (2005) 631–642.

- [20] J. Xue, Q. Ye, Computing exponentials of essentially non-negative matrices entrywise to high relative accuracy, *Math. Comp.* 82 (283) (2013) 1577–1596. doi:10.1090/S0025-5718-2013-02677-4.
URL <http://dx.doi.org/10.1090/S0025-5718-2013-02677-4>
- [21] J. Xue, Q. Ye, Entrywise relative perturbation bounds for exponentials of essentially non-negative matrices, *Numer. Math.* 110 (3) (2008) 393–403. doi:10.1007/s00211-008-0167-5.
URL <http://dx.doi.org/10.1007/s00211-008-0167-5>